Institut National des Langues et Civilisations Orientales
Département : TIM
Master en Ingénierie Multilingue
Année 2007/2008
Mémoire de recherche

La représentation du turc en Unitex

Étudiant :
Arianna Bisazza
Encadrant :
Prof. Pierre Zweigenbaum
Date de soutenance :
24 octobre 2008

Résumé

Cette étude présente un modèle pour la représentation du turc dans la plate-forme de traitement de corpus Unitex. Le choix du modèle est déterminé principalement par la morphologie agglutinante du turc et s'appuie sur une étude des outils linguistiques existants. La solution proposée inclut l'emploi de l'analyseur morphologique Zemberek. La liaison entre l'analyse du texte turc produite par Zemberek et les formalismes reconnus par Unitex est assurée par le package Java implémenté. Des exemples d'application des fonctionnalités d'Unitex au turc sont donnés en guise d'évaluation.

Mots-clés Turc - Unitex - agglutination - analyse morphologique - suffixes subordonnants

Abstract

This study presents a model for the representation of Turkish in the corpus processing platform Unitex. The choice of the model is mainly determined by the agglutinative morphology of Turkish and it relies on a study of the existing language processing tools. The solution proposed makes use of the morphological analyzer Zemberek. The link between Zemberek-produced analysis of Turkish text and Unitex-recognized formalisms is carried out by the implemented Java package. Examples of Unitex functionalities applied on Turkish are provided by way of evaluation.

Keywords Turkish - Unitex - agglutination - morphological analysis - subordinating suffixes

Table des matières

Introduction

Unitex [Paum 06],[Silb 94] est une plate-forme multilingue de traitement de corpus réunissant un ensemble de logiciels et de ressources linguistiques. Le turc ne faisant actuellement partie des langues de travail d'Unitex, l'objectif de notre travail est pallier ce manque.

Adapter Unitex au turc signifie rendre accessible un instrument d'étude de cette langue à un public non seulement d'ingénieurs linguistes, mais aussi de linguistes et d'apprenants, ce qui est démontré par la grande variété d'applications de ce logiciel.

Le choix d'une forme de représentation linguistique optimale, qui tienne compte à la fois des ressources linguistiques existantes et des contraintes imposées par les formalismes qu'Unitex implémente, est l'occasion de mener une réflexion sur la langue turque et ses spécificités, parmi lesquelles prime la morphologie de type agglutinant.

Le mémoire est structuré comme suit : les deux premiers chapitres présentent la langue turque selon une perspective d'ingénierie linguistique. Le troisième introduit Unitex et ses formalismes de représentation linguistique, tout en exposant la procédure d'ajout d'une nouvelle langue à la plate-forme. Les chapitres quatrième et cinquième constituent le cœur de l'exposé : le choix d'une forme de représentation linguistique optimale y est discuté, suivi de l'explication détaillée des réalisations techniques. Enfin les deux derniers chapitres sont consacrés à l'évaluation du travail accompli et à l'illustration de son utilité par une application concrète, à savoir l'écriture d'une grammaire de classification des subordonnées non finies.

En l'absence d'autres citations, l'ouvrage de [Gök 05] est à considérer comme la grammaire de référence de ce travail.

Chapitre 1

Les spécificités de la langue

Le turc appartient à la famille des langues turques, ou turciques, comme le turkmène, l'azéri, le kazakh et d'autres langues d'Asie Centrale. Langues turques et langues mongoles constituent ensemble la famille des langues altaïques, à laquelle certains linguistes rattachent le coréen et même le japonais.

L'ancêtre direct du turc moderne est l'ottoman, la langue administrative de l'Empire Ottoman. Le turc moderne est né en opposition à l'ottoman lors de la réforme linguistique des années 1920-1930. Celle-ci visait principalement à remplacer l'ancienne écriture arabe par une extension de l'alphabet latin et à rejeter lexique et tournures grammaticales arabo-persanes. On essaya de réhabiliter les mots désuets d'étymologie turque et de créer des néologismes à partir de racines et suffixes provenant d'autres langues turques.

L'effet général de la réforme fut une épuration, incomplète mais quand même massive, de la langue de ses éléments arabo-persans. Le turc moderne s'est ainsi rapproché de sa matrice turcique, ce qui peut s'observer notamment en morphologie dérivationnelle. L'influence arabo-persane est encore visible dans la langue d'aujourd'hui : dans le lexique bien sûr, mais aussi dans l'emploi des conjonctions subordonnantes, la présence de quelques préfixes et les irrégularités phonologiques. Il s'agit pourtant de phénomènes limités, qui n'empêchent de traiter le turc comme une langue turcique à part entière. Voyons donc quelles sont les caractéristiques principales de cette famille linguistique :

– harmonie vocalique,
– agglutination,
– syntaxe centripète (syntagmes avec tête en position finale),
– ordre libre des constituants,

– possible élision du sujet,

Le cœur de notre travail se focalisant sur le lexique, nous ne développerons ici que les phénomènes phonologiques et morphologiques.

1.1 Harmonie vocalique et autres phénomènes phonologiques

La construction des mots en turc (flexion et dérivation) est un processus transparent, qui peut être observé en synchronie, ce qui facilite notablement l'analyse automatique. Il existe pourtant en turc des phénomènes phonologiques qui peuvent déterminer des changements dans la racine et dans le suffixe lors qu'on le rattache à celle-ci. Ces changements sont toujours déduisibles du contexte, ce qui nous permet de capturer les phénomènes phonologiques par des simples règles déterministes.

Les changements peuvent arriver *a)* dans le mot auquel on rattache un suffixe, qu'il s'agisse d'une racine simple ou d'un mot contenant d'autres suffixes, *b)* dans la première consonne du suffixe et *c)* dans les voyelles du suffixe. Cette dernière classe de changements est régie par la loi de l'harmonie vocalique traitée plus bas.

Sans souci d'exhaustivité voici quelques exemples de phénomènes phonologiques de type *(a)* :

– remplacement d'une consonne sourde en position finale par sa correspondante sonore :

*mektu**p*** + *um* -> *mektu**b**um* 'ma lettre'

– redoublement d'une consonne finale :

*ha**k*** + *ım* -> *ha**kk**ım* 'mon droit'

et quelques uns de type *(b)* :

– alternance sourde/sonore de la première consonne de quelques suffixes selon le contexte gauche :

araba + *da* -> *araba**d**a* 'dans la voiture'
uçak + *da* -> *uçak**t**a* 'dans l'avion'

– effacement de la voyelle initiale des suffixes, lorsqu'ils sont rattachés à un mot se terminant par voyelle :

çanta + ***ım*** -> *çanta**m*** 'mon sac'

La loi de l'harmonie vocalique détermine l'association des voyelles proches phonétiquement à l'intérieur d'un même mot : une fois donnée la voyelle de la première syllabe du mot, on sait à quel sous-ensemble doivent appartenir les autres voyelles du mot. Le turc comporte huit voyelles, divisées en deux classes :

– postérieures : **a, ı, o, u** ;
– antérieures : **e, i, ö, ü**.

Les règles de l'harmonie vocalique sont données à la Table 1.1.

Voyelle	Voyelles pouvant la suivre dans le même mot
a	a, ı
ı	a, ı
o	a, u
u	a, u
e	e, i
i	e, i
ö	e, ü
ü	e, ü

TAB. 1.1 – Les règles de l'harmonie vocalique

Par effet de l'harmonie vocalique, un suffixe a toujours plusieurs allomorphes[1], précisément deux ou quatre selon le groupe auquel il appartient :

– dans les suffixes de 'type I' les voyelles sont hautes et varient entre 'i', 'ı', 'ü' et 'u'. Par exemple le suffixe du verbe 'être' au présent (3e personne du singulier) peut apparaître sous quatre formes :

  *pazar**dır*** 'il est dimanche'
  *perşembe**dir*** 'il est jeudi'
  *mutlu**dur*** 'il est content'
  *üzgün**dür*** 'il est triste'

– dans les suffixes de 'type A' les voyelles sont basses et non arrondies et varient entre 'a' et 'e'. C'est le cas du pluriel :

  *elma**lar*** 'pommes'
  *erik**ler*** 'prunes'

Les suffixes turcs n'ont donc pas de forme canonique à proprement parler, et l'on trouve en littérature différentes façons de les désigner isolément. Une convention très répandue consiste à noter la voyelle sujette à alternance par une lettre majuscule, symbole d'un sous-ensemble de voyelles (*-dIr* pour l'alternance {*-dir,-dır,-dür,-dur*}, *-lAr* pour {*-lar,-ler*}, *-DIK* pour {*-dIk, -dIğ, -tIk, -tIğ*} etc.) [Gök 05]. Il est autrement possible de choisir comme standard un mot quelconque (ex. 'şey')

[1]Font exception les suffixes d'origine étrangère (par ex. *-izm*), ainsi que quelques suffixes natifs comme *-(y)A**b**il, -(y)Iv**e**r, -(I)y**o**r, -(y)k**e**n*.

et de donner comme forme canonique des suffixes la forme qu'ils auraient si rattachés à ce mot (ex. -dir, -ler) [Bazi 87]. Dans ce mémoire nous utiliserons la première convention.

Le système de règles décrivant les différentes formes de surface qu'un morphème peut acquérir selon son contexte phonologique est dit morphophonémique.

1.2 Morphologie agglutinante

Les langues agglutinantes sont caractérisées par une morphologie très riche faisant large emploi d'affixes. Elles s'opposent aux langues isolantes dans lesquelles les mots sont invariables, mais se distinguent aussi des langues flexionnelles[2].

Le vocabulaire d'une langue agglutinante est formé par l'assemblage d'éléments basiques, dits affixes. Les affixes diffèrent des désinences des langues flexionnelles en ce qu'ils sont facilement discernables du radical et que chacun d'entre eux n'exprime qu'un seul trait grammatical ou sémantique. La concaténation d'affixes en langue agglutinante peut même devenir récursive[3].

Voici un exemple opposant la construction d'un mot turc à celle d'un mot français, langue flexionnelle :

Turc. À la racine, qui existe à la forme nue, on ajoute autant de morphèmes qu'il y a de traits grammaticaux à exprimer.

> ***bileceksin*** '[tu] sauras' :
> *bil* 'sache' +[Futur]+[2sing] -> *bil+**ecek**+**sin***

Français. À la racine, qui peut ne pas exister à la forme nue, on ajoute un seul morphème pour plusieurs traits grammaticaux.

---

[2] Voici une classification non exhaustive des langues selon des critères morphologiques :

- langues isolantes : thaï, vietnamien, laotien et autres langues de l'Asie du Sud-Est ;
- langues agglutinantes : langues turques, basque, coréen, finnois, japonais, zoulou ;
- langues flexionnelles : toutes les langues indo-européennes, mais à des degrés divers. Les anciennes le sont à plein titre (latin, grec ancien, sanskrit), les modernes moins (français, grec moderne, hindi). Les langues sémitiques sont aussi flexionnelles.

[3] Le suffixe pronominal *-ki*, par exemple, peut se combiner avec une forme locative ou génitive contenant déjà un suffixe *-ki* :

> *mutfağında**ki**nda**ki*** 'celui qui se trouve dans celui qui se trouve dans la cuisine'
> *kedinin**ki**nde**ki*** 'celui qui se trouve dans celui du chat'

De telles accumulations de suffixes sont évidemment rares, bien que grammaticalement correctes.

sauras :
saur + [Futur]+[2sing] -> saur+**as**

L'agglutination en turc s'effectue toujours par le biais de suffixes et concerne non seulement la flexion, mais aussi la dérivation lexicale.

sakinleştirebil(mek)[4] 'pouvoir calmer'
sakin 'calme'
*sakin**leş**-* 'devenir calme', 'se calmer'
*sakinleş**tir**-* 'faire devenir calme', 'calmer'
*sakinleştir**ebil**-* 'pouvoir faire devenir calme', 'pouvoir calmer'

bağlılık 'fidélité'
bağ 'lien'
*bağ**lı*** 'lié'
*bağlı**lık*** 'attachement', 'fidélité'

Le turc étant originairement dépourvu de conjonctions[5], des éléments complexes de la langue, tels la coordination et la subordination, peuvent, eux aussi, être exprimés par le biais de suffixes.

*Dün sinemaya gid**ip** bir film seyrettik.*
'Hier nous sommes allés au cinéma **et** nous avons regardé un film.'

*Yusuf eve gel**diğinde** bir film seyrettik.*
'**Quand** Yusuf est rentré à la maison, nous avons regardé un film.'

Le système de règles déterminant quels suffixes peuvent s'attacher à une racine donnée et dans quel ordre est dit morphotactique. Par exemple, le paradigme de flexion nominale (Table 1.2) établit qu'une racine nominale peut être suivie d'un ou zéro suffixes de nombre, puis d'un ou zéro suffixes de possession, puis d'un ou zéro suffixes de cas. La morphotactique des suffixes turcs

	RACINE	+ (NOMBRE)	+ (POSSESSION)	+ (CAS)	
ex.	*ev*	*-ler*	*-im*	*-de*	-> *evlerimde*
					'dans mes maisons'

TAB. 1.2 – Paradigme de flexion nominale

constitue un langage régulier. Comme nous le verrons par la suite, l'analyse morphologique peut donc être effectuée par des automates munis d'un dictionnaire de racines et de suffixes.

[4] *-mAk* est la marque de l'infinitif. En tant que suffixe flexionnel, elle suit les suffixes dérivationnels.
[5] Les conjonctions employées en turc moderne ont été empruntées principalement au persan et à l'arabe.

Il est important de remarquer que, par l'effet des propriétés mentionnées, le mot (au sens de mot graphique) n'a pas la même portée dans les langues agglutinantes et dans les langues flexionnelles. Un mot turc peut, en effet, englober un grand nombre d'éléments linguistiques qui dans les langues indo-européennes constituent des mots indépendants (comme les adjectifs possessifs, les prépositions, la négation, les modificateurs verbaux etc.).

Kitaplarımı okumanı istemiyorum.
'Je ne veux pas que tu lises mes livres.'

kitaplarımı 'mes livres(acc.)'
okumanı '[que] tu lises'
istemiyorum '[je] ne veux pas'

La portée d'un mot turc peut ainsi varier entre le mot (au sens français du terme), le syntagme nominal ou prépositionnel, jusqu'à correspondre à une proposition entière. Cela explique une grande part des difficultés rencontrées dans la représentation du lexique turc au moyen du formalisme DELAF, et apporte un argument supplémentaire à la « solution C » présentée plus loin.

Chapitre 2

Analyse morphologique automatique du turc

L'étiquetage morphosyntaxique en TAL est un traitement fondamental du texte, étape obligée de la plupart des applications d'ingénierie linguistique. Cela consiste à repérer la forme canonique (lemme) d'un mot et en expliciter les traits morphosyntaxiques.

L'analyse morphologique se distingue de l'étiquetage morphosyntaxique en ce qu'elle n'inclut aucune opération de désambiguïsation, mais consiste uniquement à donner toutes les analyses possibles d'un mot, après l'avoir décomposé en unités minimales porteuses de sens (morphèmes).

disent	-> di(s) + ent	-> dire + [Pres] + [3plur]
söylüyorlar	-> *söyl(ü)* + *yor* + *lar*	-> *söyle* + [Pres] + [3plur]
personnes	-> personne + s	-> personne + [Plur]
kişiler	-> *kişi* + *ler*	-> *kişi* + [Plur]

Dans des langues à la morphologie relativement pauvre, comme l'anglais et le français, l'étiquetage morphosyntaxique est généralement réalisé par l'application au texte d'un dictionnaire de formes fléchies préalablement catégorisées (cf. DELAF pour le français) suivie d'une phase de désambiguïsation par règles ou statistique. Par conséquent, une vraie analyse morphologique se révèle nécessaire uniquement pour les mots inconnus (non recensés par le dictionnaire).

Cette méthode n'est pas envisageable en turc, comme dans les autres langues agglutinantes, car les combinaisons possibles de suffixes étant pratiquement illimitées, le lexique des formes fléchies aurait des dimensions énormes sans jamais assurer une couverture satisfaisante de la langue.

L'agglutination rend donc essentielle la mise au point d'un bon analyseur morphologique.

Nous allons maintenant présenter un état de l'art des analyseurs morphologiques existants pour le turc.

2.1 PC-Kimmo

PC-Kimmo [Antw 93] est un analyseur basé sur les transducteurs finis, qui implémente le formalisme des règles à deux niveaux. Ce modèle morphologique, conçu à l'origine pour la langue finnoise par Kimmo Koskenniemi [Kosk 83], suit la division traditionnelle faite par les linguistes entre morphophonémique et morphotactique : la première est stockée séparément dans une base de règles implémentées par des transducteurs, tandis que la seconde est capturée par le lexique contenant la forme lexicale des racines et affixes, ainsi que les contraintes morphotactiques. La seconde composante est implémentée par des automates finis.

La particularité des règles à deux niveaux est l'alphabet sur lequel elles sont définies, à savoir un ensemble de couples dites *couples de correspondance* dont le premier élément est un symbole sous-jacent (ou lexical), le second un symbole de surface. Tout mot est vu au niveau sous-jacent comme une suite de morphèmes (ex. 'personne+s'), et au niveau de surface comme réalisation concrète suivant les règles phonologiques de la langue (ex. 'personnes'). Par exemple, le mot *çantam* 'mon sac', composé de *çanta* 'sac' plus le suffixe possessif *-(I)m* 'mon', est représenté comme suit :

Forme lexicale :	ç	a	n	t	a	+	I	m
Forme de surface :	ç	a	n	t	a	0	0	m

Le symbole '0' représente la chaîne vide et '+' la frontière d'un morphème.

La notation d'un couple de correspondance est la suivante :

'<symbole sous-jacent> :<symbole de surface>'
par exemple 'A :a', 'p :b', 'I :0'

Ci-dessous est reportée la règle qui établit l'effacement de la voyelle initiale d'un suffixe lorsqu'il est précédé d'une autre voyelle (cette règle s'applique au mot *çantam*). Dans ce cas seul le contexte gauche est renseigné, ce qui suit la voyelle n'étant pas pertinent dans cette règle.

Les règles à deux niveaux diffèrent des règles de réécriture de la grammaire générative sous plusieurs aspects :

– elles s'appliquent en parallèle et non séquentiellement,

Couple	Opérateur	Contexte gauche	Contexte droit
I :0	<=>	VOWEL :VOWEL (' :') + :0 _	

– il n'y a pas de niveaux intermédiaires de dérivation, mais uniquement deux niveaux de représentation,
– la relation entre niveau lexical et de surface est préservée tout au long du processus d'analyse par l'emploi des couples de correspondance, ce qui rend indifférent l'ordre d'application des règles,
– les règles à deux niveaux sont bi-directionnelles : le même ensemble de règles peut être si bien employé en reconnaissance qu'en génération.

Des descriptions morphologiques à deux niveaux pour le turc compatibles avec le moteur d'analyse de PC-Kimmo ont été proposées par Kemal Oflazer en 1993 [Ofla 93] et Serdar Murat Öztaner en 1996 [Özt 96]. Nous avons testé la première sous la distribution nommée Turklex qui date de 1994. Les exécutables de PC-Kimmo fournis sur le site officiel du programme fonctionnent correctement sous Windows, par contre nous n'avons pas réussi à compiler les sources.

2.2 Zemberek

Zemberek [Akı] est une bibliothèque Java *open source* de traitement de texte spécialement conçue pour le turc et les autres langues turciques (azéri, tatare et turkmène sont incluses dans la distribution actuelle). Cette bibliothèque assure les fonctions de contrôle orthographique dans la version courante d'OpenOffice pour le turc. Les opérations réalisables par Zemberek incluent analyse et génération morphologique.

La libraire est organisée en deux parties : d'un côté l'information linguistique propre à chaque langue, de l'autre les fonctions de traitement de texte. La Fig. 2.1 montre comment l'information linguistique est organisée dans Zemberek. Le lexique des racines, constituant la plus grande source de données (700 KB pour le turc), est stocké sous forme d'arbre lexicographique afin de diminuer le coût de recherche. Le lexique des suffixes, par contre, est codé au format XML. Ce dernier contient aussi les contraintes morphotactiques de chaque suffixe, semblablement à ce qu'on a vu dans PC-Kimmo.

En revanche la morphophonémique ne constitue pas un module en soi dans Zemberek, mais se trouve partagée entre trois composantes : les deux lexiques et un mécanisme appelé *produc-*

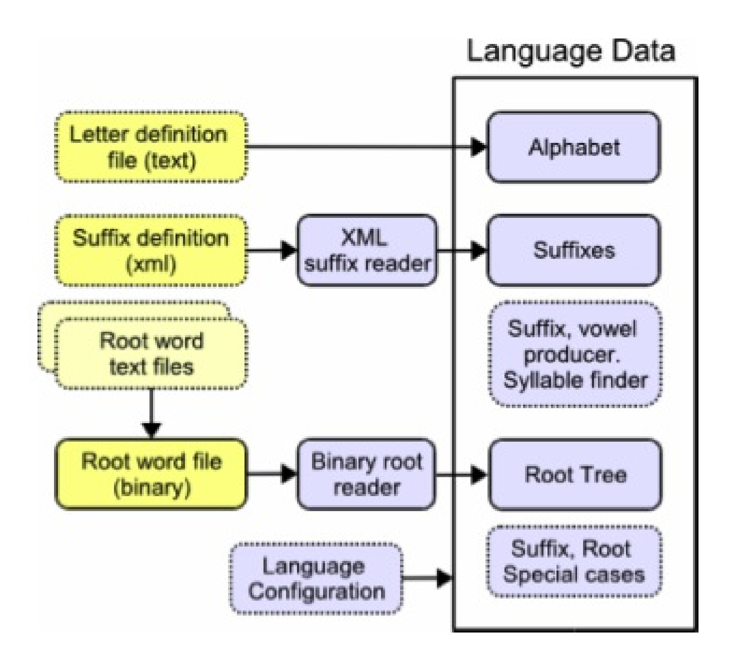


FIG. 2.1 – L'information linguistique dans Zemberek

teur de suffixes. L'arbre des racines contient pour chaque racine toutes ses réalisations possibles en présence d'un suffixe ; les insertions de consonnes avant certains suffixes, ainsi que d'autres phénomènes phonologiques n'affectant pas la racine, sont définies pour chaque suffixe dans le lexique de suffixes même ; enfin l'harmonie vocalique est gérée par la classe *producteur de suffixes*.

Étant donné un mot *M* en entrée, l'algorithme d'analyse est le suivant :

- calculer les racines candidates de *M* (tous les mots du lexique des racines étant un préfixe de *M*),
- pour chaque racine candidate, y ajouter les suffixes admis par les règles morphotactiques et y appliquer les transformations phonologiques pertinentes, jusqu'à ce qu'on arrive à reconstruire *M* ou jusqu'à ce qu'il ne soit plus possible d'ajouter de suffixes.

Des exemples d'analyse produits par Zemberek sont donnés à l'Annexe A.

Les différentes fonctions de Zemberek ont été d'abord testées en utilisant la classe de test `DemoMain` fournie dans la distribution téléchargeable sur le site du projet[1], puis en développant une classe de test ad hoc. Au point de vue logiciel, la bibliothèque est bien construite, accessible, et facilement intégrable dans de nouvelles applications. La documentation est très réduite et presque

[1]http ://code.google.com/p/zemberek/

uniquement en turc.

Le côté linguistique au contraire paraît moins soigné : le lexique de suffixes devrait être en cours d'amélioration par les développeurs. La terminologie utilisée pour les suffixes n'est pas fixée, et on rencontre même des étrangetés, comme le suffixe nommé *ISIM_TAMLAMA_I* traité en détail à la Section 5.2. Il est heureusement possible de rectifier le lexique des suffixes en modifiant le fichier XML le contenant. En revanche le lexique des racines est donné uniquement sous forme de fichier binaire. Il est tout de même possible de créer ses propres dictionnaires et de les ajouter à la base de racines existante.

L'analyseur de Zemberek, de même que PC-Kimmo, implémente un algorithme *root driven* (analyse gauche-droite, ou guidé par la racine), par conséquent il ne peut pas prévoir la catégorie d'un mot inconnu.

2.3 L'analyseur *affix-stripping* d'Eryiğit

Un autre analyseur a récemment été développé pour le turc par les chercheurs Gülşen Eryiğit et Eşref Adalı [Eryi 04].

Tandis que les deux précédents analyseurs suivent une approche de type *root driven*, celui-ci utilise l'approche inverse, dite *affix stripping* (analyse droite-gauche). Celle-ci consiste à chercher à reconnaître et à retirer les suffixes en partant de la fin de la chaîne, jusqu'à ce qu'aucun suffixe ne soit plus reconnu. La partie restante est donc élue racine. Un lexique de racines peut être utilisé pour confirmer cette hypothèse, mais n'est pas requis par l'algorithme, ce qui représente un grand avantage pour les langues dépourvues de ressources de TAL. Ce genre d'algorithme peut en outre servir à prévoir la catégorie des mots inconnus du dictionnaire (néologismes, noms propres etc.) présents dans toute les langues. En contrepartie, cet algorithme, lorsqu'il est employé sans dictionnaire, a le défaut de ne pas savoir repérer le véritable lemme de certaines formes ambiguës. En effet tous les suffixes, qu'ils soient flexionnels ou dérivationnels, sont retirés jusqu'à ce que la racine soit complètement dénudée. Ainsi un mot comme *günlük* ne recevrait qu'une seule de ses deux correctes analyses :

günlük	-> gün + lük	-> gün + [Adj] ('du jour, journalier')
~~günlük~~	-> ~~günlük~~	-> ~~günlük ('encens')~~

L'analyseur d'Eryiğit est entièrement basé sur des automates finis. Les règles morphotactiques

sont appliquées à l'inverse (la chaîne est parcourue de droite à gauche), ce qui implique l'inversion des FSA mêmes.

Au point de vue linguistique, le travail d'Eryiğit et Adalı présente une intéressante classification des suffixes selon leur fonction et la catégorie des racines auxquelles ils peuvent être ajoutés. Alors que la morphotactique constitue un seul paquet de règles dans Zemberek, ici chaque catégorie de suffixes est traitée par un FSA différent. Le travail de transformation des automates est aussi bien documenté par l'article.

L'analyseur d'Eryiğit n'a pas pu être testé car le site renseigné par l'article n'était pas accessible au moment de la rédaction de ce mémoire.

PC-Kimmo reste *le* travail de référence dans la littérature du domaine de l'analyse morphologique du turc. Cependant, il s'adapte mal aux besoins de notre projet. Il n'est accessible gratuitement que dans une implémentation très ancienne. Le lexique des racines est aussi limité.

L'analyseur d'Eryiğit semble être le fruit d'un travail méthodique réunissant points forts computationnels et linguistiques. Malheureusement son inaccessibilité au moment de la rédaction de ce mémoire nous a empêché de le choisir comme la ressource à intégrer dans Unitex.

Zemberek, quoique moins bien documenté et peut-être moins scientifiquement fondé dans sa partie linguistique, est à notre connaissance la ressource du domaine la plus accessible et facilement réutilisable pour le développement d'une nouvelle application, en tant que bibliothèque *open source*. Pour ces raisons nous l'avons choisi comme notre outil linguistique de travail (cf. Chapitre 4). On peut espérer que les problèmes rencontrés dans sa distribution actuelle seront réglés prochainement par les développeurs.

Chapitre 3

Représenter une langue en Unitex

Unitex [Paum 06],[Silb 94] est une plate-forme multilingue de traitement de corpus réunissant un ensemble de logiciels et de ressources linguistiques. L'interface graphique du programme a été développée par le Laboratoire d'informatique de l'IGM (Université de Marne-la-Vallée). Les ressources linguistiques fournies pour le français sont issues de travaux initiés par Maurice Gross au Laboratoire d'Automatique Documentaire et Linguistique (LADL). Ces travaux ont été étendus à d'autres langues au travers du réseau de laboratoires RELEX.

Ce chapitre présente les ressources requises par Unitex pour travailler sur une langue donnée, ainsi que quelques expériences effectuées sur des langues posant, comme le turc, des problèmes complexes.

3.1 Les ressources à fournir

Afin de pouvoir charger un corpus sur Unitex et d'y effectuer des recherches de motifs simples, seuls les fichiers de l'alphabet et de l'ordre de tri sont indispensables. Les graphes de prétraitement peuvent être empruntés, et éventuellement modifiés si les conventions de la langue le requièrent, aux langues déjà présentes dans Unitex. Il est en outre souhaitable de fournir un corpus de test dans le répertoire de la nouvelle langue.

L'intérêt d'Unitex réside pourtant dans la possibilité d'appliquer aux textes des connaissances (surtout morphologiques, mais aussi syntaxiques, sémantiques etc.) qui sont spécifiques à la langue traitée. Le vrai travail consiste donc à construire les dictionnaires électroniques, et à éventuellement écrire des grammaires et des tables de lexique-grammaire. La liste exacte des ressources à fournir n'est donc pas fixée à l'avance mais dépend de ce que l'on veut faire d'Unitex.

Donner à Unitex un moyen pour effectuer l'analyse morphologique des mots (sous la forme du dictionnaire des formes fléchies ou selon d'autres techniques) constitue en tout cas le point de départ obligé pour rendre le logiciel réellement opérationnel.

3.1.1 Dictionnaires : le formalisme DELA

Les dictionnaires électroniques constituent le premier moyen d'enrichir le texte d'un certain nombre de connaissances linguistiques. Dans les langues non agglutinantes le dictionnaire de formes fléchies est la ressource fondamentale qui permet l'analyse morphologique des mots (cf. Chapitre 2). Chaque entrée contient trois champs obligatoires, à savoir la forme fléchie, la forme canonique (le lemme, selon les conventions de la langue) et la séquence d'informations grammaticales et sémantiques du mot. Le deuxième champ peut être renseigné par la chaîne vide lorsque le lemme coïncide avec la forme fléchie. Le dernier champ, optionnel, représente la séquence d'informations flexionnelles. Selon la langue, ces informations peuvent indiquer le genre, le nombre, le temps et mode de conjugaisons, la déclinaison etc. Chaque code flexionnel doit être constitué d'un seul caractère, ce qui limite quelque peu le pouvoir d'expression de ce champ.

```
fiches,fiche.N+z1:fp
fiches,fiche.V+z1:P2s:S2s
fiches,ficher.V+z1:P2s:S2s
```

Il est important de remarquer que les deux séquences d'informations (3e et 4e champs) ne s'interprètent pas de la même façon : alors que le plus (+) séparant les codes sémantiques-grammaticaux est lu comme un *ET* logique, les deux points (:) représentent un *OU* logique, les membres de la disjonction étant des ensembles de codes flexionnels, eux-mêmes non ordonnés. Dans l'exemple ci-dessous la séquence ":P2s:S2s" signifie donc *"présent 2e personne du singulier OU subjonctif 2e personne du singulier"*.

Le dictionnaire des formes fléchies est généralement accompagné de dictionnaires de formes simples. Ceux-ci contiennent les formes canoniques des mots suivies d'une description de la catégorie grammaticale et d'éventuels codes de flexion, ainsi que d'informations sémantiques diverses. Le format de ces dictionnaires (DELAS) diffère du DELAF seulement en ce que seules les formes canoniques des mots y sont renseignées, ce qui réduit à deux le nombre de champs obligatoires :

```
cheval,N4+Anl
```

Parmi ces deux formalismes, seul le DELAF a été utilisé dans le cadre de ce travail, les dictionnaires de formes fléchies produits contenant à eux seuls toutes les informations linguistiques renseignées (cf. Section 4.2).

3.1.2 Grammaires

Au point de vue formel les grammaires d'Unitex sont des grammaires algébriques étendues intégrant la notion de transduction. Elles sont donc utilisées pour reconnaître des motifs, mais aussi pour produire des sorties qui vont remplacer ou s'ajouter au texte analysé. Ce formalisme se prête à une représentation graphique conviviale implémentée par l'éditeur de graphes de l'interface Unitex.

Parmi les nombreuses utilisations des graphes, mentionnons ici la normalisation de formes ambiguës (dont le résultat est visualisable au moyen de l'automate du texte), les graphes syntaxiques, la recherche de motifs et la levée d'ambiguïtés à travers le formalisme ELAG [Lapo 98].

L'automate du texte est un moyen efficace de visualiser les ambiguïtés lexicales contenues dans une phrase. Chaque phrase du texte est en effet représentée par un automate dont les chemins expriment les diverses interprétations possibles. Les graphes de normalisation permettent de normaliser des formes ambiguës. Si une même forme admet plusieurs analyses, elles seront toutes insérées dans l'automate du texte. Un exemple de graphe de ce genre est constitué par la normalisation de l'élision en français contenue dans le répertoire d'Unitex pour le français.

Les autres utilisations des graphes seront reprises au Chapitre 7.

3.1.3 Lexiques-grammaire

Les tables de lexique-grammaire [Gros 86] servent à représenter les propriétés syntaxiques des éléments d'une langue. Parmi leurs applications, prime la désambiguïsation lexicale. Unitex intègre les tables des verbes français développées par Maurice Gross et son équipe du LADL pour décrire le nombre et la nature des compléments admis par le verbe et les différentes transformations que ce verbe peut subir. Le formalisme du lexique-grammaire n'est pas cantonné aux verbes, mais peut également être appliqué à la description d'autres éléments lexicaux de la langue.

Il existe des lexiques-grammaires dans des langues très diverses (allemand, italien, coréen, malgache etc.). La construction de lexiques-grammaire pour le turc ne rentre pas dans les objectifs de ce projet. Il s'agirait pourtant d'une ressource très précieuse dans le traitement de cette langue.

3.2 Expériences sur d'autres langues

La dernière distribution d'Unitex, conçu originairement pour le français, contient des ressources (dictionnaires et graphes de prétraitement principalement) spécifiques à plusieurs autres

langues européennes, parmi lesquelles figurent anglais, italien, allemand, anglais, grec, norvégien etc. Il est également possible d'utiliser des ressources externes après les avoir converties au format DELAF.

Unitex a aussi été employé dans des projets concernant des langues à la morphologie complexe, dont le traitement requiert des stratégies alternatives à celles couramment utilisées pour le français.

3.2.1 Finnois

Le travail de Marie Calberg [Calb 03] propose un modèle pour le traitement de la morphologie du finnois, langue agglutinante comme le turc. Le modèle est fondé sur des transducteurs à nombre fini d'états réalisés sous forme de graphes qui ont été dessinés en utilisant le module graphique d'INTEX/UNITEX. L'analyse morphologique du finnois est effectuée par des graphes lexicaux, c'est-à-dire des graphes dont les états sont des caractères et non pas des tokens, décrivant chacun un morphème. Après la détection des différents morphèmes, des règles de linéarisation (représentées elles aussi par des graphes) sont appliquées à la sortie des graphes morphémiques pour produire les formes de surface valides. Cette approche est en partie inspirée de la morphologie à deux niveaux [Kosk 83] mais n'implémente pas le même formalisme. La stratégie de Calberg peut être définie comme générative, dans la mesure où le lexique est généré pour et seulement pour les mots du texte que l'on veut traiter. Elle peut donc bien s'adapter au traitement du turc.

3.2.2 Thaï

Les langues extrême-orientales, quant à elles, soulèvent le problème de la segmentation en mots encore avant celui de l'analyse morphologique. Dans le traitement de cette langue, l'information morphologique (sous forme de dictionnaires de mots et de morphèmes) est utilisée pour calculer la meilleure tokénisation du texte. Le manuel d'Unitex [Paum 06] présente brièvement les opérations réalisées sur la langue thaïe : pour chaque phrase toutes les segmentations possibles sont calculées, puis visualisées à travers l'automate du texte. L'option *'Clean Text FST'* permet ensuite de ne garder que les meilleurs chemins selon le critère "moindre nombre de mots inconnus contenus".

3.2.3 Discussion

Nos propositions de représentation du turc dans Unitex sont en partie inspirées des deux travaux présentés ci-dessus. Au premier, nous reprenons l'approche générative du lexique. Au second,

l'idée de représenter les différentes segmentations possibles d'une phrase à travers l'automate du texte. Si le lien avec le finnois est évident, les deux langues étant agglutinantes, le lien avec le thaï l'est moins. En fait l'analyse morphologique d'un mot turc peut aussi être considérée comme un problème de segmentation, en une racine plus une séquence de suffixes. Nous pouvons donc envisager la réalisation d'un automate du texte dont les états seraient non pas les mots, mais les morphèmes, ce qui nous permettrait de visualiser et éventuellement traiter les ambiguïtés morphologiques de la langue.

Chapitre 4

La représentation du turc en Unitex

Les problèmes soulevés par la représentation du turc en Unitex s'articulent autour de deux questions principales :

– comment exploiter les ressources linguistiques existantes et à quel niveau relier l'analyseur morphologique choisi à Unitex ?
– comment adapter le formalisme des dictionnaires DELAF à la représentation d'une morphologie de type agglutinant ?

Les deux sections qui suivent répondent respectivement à ces questions.

4.1 Comment exploiter les ressources existantes

Nous voulons maintenant transférer dans Unitex l'information codée par l'analyseur morphologique que nous avons choisi comme ressource linguistique de départ. Il existe en gros deux façons de le faire :

(a) l'une consiste à transformer en dictionnaire de formes simples le lexique des racines de l'analyseur, puis à convertir en graphes lexicaux les règles morphologiques et phonologiques implémentées par l'analyseur. Le dictionnaire des formes fléchies est donc produit dans Unitex par l'application des graphes lexicaux aux formes du texte ;
(b) l'autre revient à convertir en dictionnaire DELAF la sortie de l'analyseur lancé sur le texte que l'on veut charger sur Unitex.

Dans les deux cas on produira un dictionnaire de formes fléchies spécifique au texte ne décrivant que les formes contenues dans celui-ci.

La première solution (a) est probablement la plus « propre » au point de vue formel. Elle est néanmoins plus coûteuse, parce qu'elle demande la construction d'un nouveau dispositif d'analyse, et non optimale sur le plan technique, puisqu'elle nous empêche de profiter des optimisations logicielles mises au point par les développeurs de l'analyseur. De plus, dans le cas de Zemberek, les contraintes morphotactiques et morphophonémiques ne sont exprimées ni sous formes de règles génératives, ni de règles à deux niveaux. Les convertir en transducteurs ne serait donc pas immédiat. En ce qui concerne PC-Kimmo aussi, des difficultés de portabilité du modèle morphologique à deux niveaux à des outils non basés sur ce formalisme ont été pointées par [Aleg 08] dans l'adaptation à la langue basque d'un correcteur orthographique *open source*. En dehors de la morphologie à deux niveaux, tous les modèles pour le traitement du turc que nous connaissons requièrent une analyse en plusieurs passes (au moins deux : morphotactique et morphophonémique), ce qui n'est pas, à ce jour, facilement réalisable dans Unitex de façon automatique[1].

Pour ces raisons la seconde solution (b) nous paraît plus sensée. Selon notre proposition de représentation du turc en Unitex, l'analyse morphologique d'un texte se fera en dehors de la plateforme, avant que le texte puisse être chargé dans celle-ci. Après l'avoir implémentée, nous pouvons affirmer que cette solution ne représente pas de coûts supplémentaires conséquents, grâce aussi à la performance de l'analyseur employé.

Enfin, si la ressource dont nous avons besoin existe déjà et fonctionne, la reproduire en utilisant un autre formalisme serait, à notre avis, un exercice stérile. Notre objectif n'est donc pas l'implémentation d'un compilateur de règles morphologiques, ni la conversion de règles existantes en graphes compilables par Unitex, mais plutôt l'exploitation en Unitex des informations produites par un analyseur morphologique existant. Nous allons dorénavant travailler sur l'analyse produite par Zemberek sans nous occuper en détail de la façon dont il fonctionne. Nous envisageons pourtant une phase de correction automatique minimale de la sortie de l'analyseur, afin d'éliminer les erreurs les plus répandues (cf. Section 5.2). Les développements futurs de Zemberek, logiciel libre, pourraient rendre cette phase de correction superflue.

---

[1] « [...] l'application récursive des graphes morphologiques indépendants n'a pas encore été vraiment implantée dans cette plateforme [Unitex]. Par exemple, il n'est pas possible de décomposer dynamiquement des entrées complexes pour produire une analyse morpho-sémantique. » [Calb 03].

4.2 Le turc dans le formalisme DELAF

Quelle que soit la solution choisie, l'information morphologique extraite du texte doit, au final, être codée dans un formalisme reconnu par Unitex. Pour atteindre cet objectif, il nous faut réfléchir à plusieurs questions : que représente le lemme (ou forme canonique) d'une entrée DELAF[2] ? Quelles informations doivent véhiculer les codes sémantiques-grammaticaux et quelles autres les codes flexionnels ? Comment gérer la forme de surface et la forme sous-jacente d'un mot ?

Nous allons maintenant exposer trois différentes façons d'adapter la syntaxe du DELAF à la représentation des formes turques. La meilleure solution sera celle qui conserve un maximum d'information morphologique produite par l'analyseur tout en étant compatible avec un maximum de fonctionnalités d'Unitex.

4.2.1 Solution A : le lemme est la forme lexicale

Une première solution consiste à coder comme lemme de l'entrée DELAF la forme lexicale (sous-jacente) tout entière, c'est-à-dire la racine suivie d'une séquence éventuellement vide de suffixes, eux-aussi à la forme lexicale, séparés par un caractère spécial. Les formes *adaletin* 'ta justice', 'de la justice' et *çalıştılar* 'travaillèrent', 's'entre-volèrent' seraient donc décrites par les entrées suivantes[3] :

```
adaletin,adalet#In.N:2
adaletin,adalet#In.N:g
...
çalıştılar,çalış#tI#lAr.V:P6
çalıştı,çal#Iş#tI#lAr.V+Rec:P6
```

Ceci est le meilleur moyen de coder la segmentation du mot en morphèmes. Par contre, de cette façon, il est très malaisé d'exprimer les informations grammaticales et sémantiques contenues par chacun de ces morphèmes, surtout lorsque le nombre de suffixes est grand. Les codes sémantiques-grammaticaux réfèrent à la forme fléchie tout entière, alors que dans notre cas on voudrait pouvoir associer une ou plusieurs informations à chaque morphème. Un autre problème se pose : la racine du mot n'est pas directement accessible, mais doit être recalculée à partir du

[2]Dans la présente section les termes 'lemme' et 'forme canonique' ne sont pas employés au sens commun, mais désignent uniquement le deuxième champ de l'entrée DELAF.

[3]Les codes flexionnels employés dans cet exemple signifient respectivement : **N** nom, **V** verbe, **Rec** voix réciproque, **P** passé, **6** marque verbale de la 3e personne du pluriel, **g** génitif, **2** possessif 2e pers. sing.

lemme par soustraction des suffixes[4]. Cette solution rend donc impossible la recherche de motifs sur la racine.

4.2.2 Solution B : le lemme est la racine

Nous pouvons autrement envisager de coder comme lemme de l'entrée DELAF la racine du mot turc. Cette solution s'écarte moins de la notion linguistique de lemme et paraît plus logique. La difficulté réside cette fois-ci dans la façon de coder les morphèmes. Les codes sémantiques-grammaticaux n'ont pas de contraintes sur la longueur et sont illimités en nombre. Nous pouvons donc choisir de noter chaque nom de suffixe comme un trait grammatical[5] :

```
>adaletin,adalet.ISIM_KOK+ISIM_TAMLAMA_IN
>adaletin,adalet.ISIM_KOK+ISIM_SAHIPLIK_SEN_IN
...
>çalıştılar,çalış.FIIL_KOK+FIIL_GECMISZAMAN_DI+FIIL_KISI_ONLAR
>çalıştılar,çal.FIIL_KOK+FIIL_BERABERLIK_IS+FIIL_GECMISZAMAN_DI
+FIIL_KISI_ONLAR
```

De cette façon, toute l'information flexionnelle est incluse dans le 3e champ. L'utilisateur est censé connaître quel trait est représenté par chaque suffixe, ce qui est seulement en partie suggéré par son nom. Recherches de motifs et grammaires devront contenir les noms des suffixes mêmes.

Il faut toutefois faire attention à la sémantique de cette représentation : selon la syntaxe du DELAF, le 3e champ est interprété comme un ensemble de traits et non pas comme une suite ordonnée. Nous perdons ainsi l'ordre d'occurrence des suffixes. Ce problème peut être contourné en faisant débuter chaque trait par son numéro d'ordre :

```
>adaletin,adalet.0ISIM_KOK+1ISIM_TAMLAMA_IN
>adaletin,adalet.0ISIM_KOK+1ISIM_SAHIPLIK_SEN_IN
...
>çalıştılar,çalış.0FIIL_KOK+1FIIL_GECMISZAMAN_DI+2FIIL_KISI_ONLAR
>çalıştılar,çal.0FIIL_KOK+1FIIL_BERABERLIK_IS+2FIIL_GECMISZAMAN_DI
+3FIIL_KISI_ONLAR
```

---

[4]Le moteur de recherche de motifs d'Unitex permet de filtrer les unités lexicales recherchées par des expressions régulières (filtres morphologiques), mais celles-ci ne s'appliquent qu'à la forme fléchie du mot. Il est possible, par exemple, d'imposer que les mots recherchés commencent par 'ab', mais non pas que leur lemme commence par 'ab'.

[5]Les noms des suffixes employés dorénavant sont ceux de l'analyseur Zemberek : *ISIM_KOK* 'racine nominale', *FIIL_KOK* 'racine verbale', *ISIM_TAMLAMA_IN* 'génitif', *ISIM_SAHIPLIK_SEN_IN* 'possessif 2e pers. sing.', *FIIL_GECMISZAMAN* 'temps verbal passé', *FIIL_KISI_ONLAR* 'accord verbal 3e pers. plur.', *FIIL_BERABERLIK_IS* 'voix verbale réciproque'. Cf. Annexe B.

Ce remède n'est pas sans conséquences : les noms de suffixe sont considérablement multipliés. De plus, en rajoutant le numéro d'ordre au nom des suffixes, on introduit une différenciation artificielle parmi des suffixes identiques qui apparaissent à des positions différentes, chose tout à fait courante. Il n'y a pas à notre connaissance une fonctionnalité native d'Unitex qui permette de factoriser les codes sémantiques-grammaticaux différant, comme dans ce cas, d'un seul caractère.

Afin de limiter la perte d'information liée à l'ordre des suffixes et de nous rapprocher encore plus de la sémantique originelle d'une entrée DELAF, nous pouvons convertir les noms des suffixes flexionnels, placés en turc après les suffixes dérivationnels dans la quasi totalité des cas, en codes flexionnels et les rentrer dans le 4e champ, ce qui a pour effet d'alléger le 3e champ et de mieux équilibrer la représentation de la forme fléchie. Les autres[6] suffixes, par contre, resteront codés comme informations sémantiques-grammaticales. Nos deux formes seraient donc représentées comme suit[7] :

```
>adaletin,adalet.ISIM_KOK:t
>adaletin,adalet.ISIM_KOK:2
...
>çalıştılar,çalış.FIIL_KOK:P6
>çalıştılar,çal.FIIL_KOK+FIIL_BERABERLIK_IS:P6
```

Cette représentation rend également plus aisées la recherche de motifs et l'écriture de grammaires.

4.2.3 Solution C : chaque morphème est une entrée du dictionnaire

La dernière solution que nous envisageons revient à considérer tous les morphèmes comme des mots à part entière ayant chacun sa propre entrée dans le dictionnaire.

```
>adalet,.ISIM
>in,.ISIM_TAMLAMA
>in,.ISIM_SAHIPLIK_SEN
...
>çalış,.FIIL
>çal,.FIIL
>ış,iş.FIIL_BERABERLIK
>tı,di.FIIL_GECMISZAMAN
>lar,onlar.FIIL_KISI
```

[6]En réalité, comme on le verra à la Section 5.1, tous les suffixes classés comme flexionnels par [Gök 05] ne seront pas convertis en codes flexionnels.

[7]Les codes flexionnels employés dans cet exemple suivent la terminologie turque de Zemberek. Ils seront présentés à la Section 5.1.

Cette représentation permet de coder autant d'informations morphologiques que l'on souhaite, mais elle présente une difficulté supplémentaire par rapport aux deux autres, due au fait qu'un mot turc peut admettre plus d'une segmentation morphologique. En effet, même si le turc est une langue à la morphologie régulière, il existe quelques suffixes homonymes très fréquents qui génèrent des ambiguïtés uniquement solubles à l'échelle syntaxique. L'exemple ci-dessous montre les trois analyses possibles du mot *masalı* :

masalı	-> masal + ı	-> masal + [Acc]	'l'histoire'
masalı	-> masal + ı	-> masal + [Poss3s]	'son histoire'
masalı	-> masa + lı	-> masa + [suff.dériv.]	'avec table', 'muni de table'

L'automate du texte (cf. Sous-section 3.1.2) est un bon moyen de visualiser les différentes segmentations possibles d'une phrase dans Unitex. La Fig. 4.1 montre une portion d'automate du texte décrivant les différentes segmentations du syntagme *ilişkiler geriştirilmesi* 'le développement des relations'.

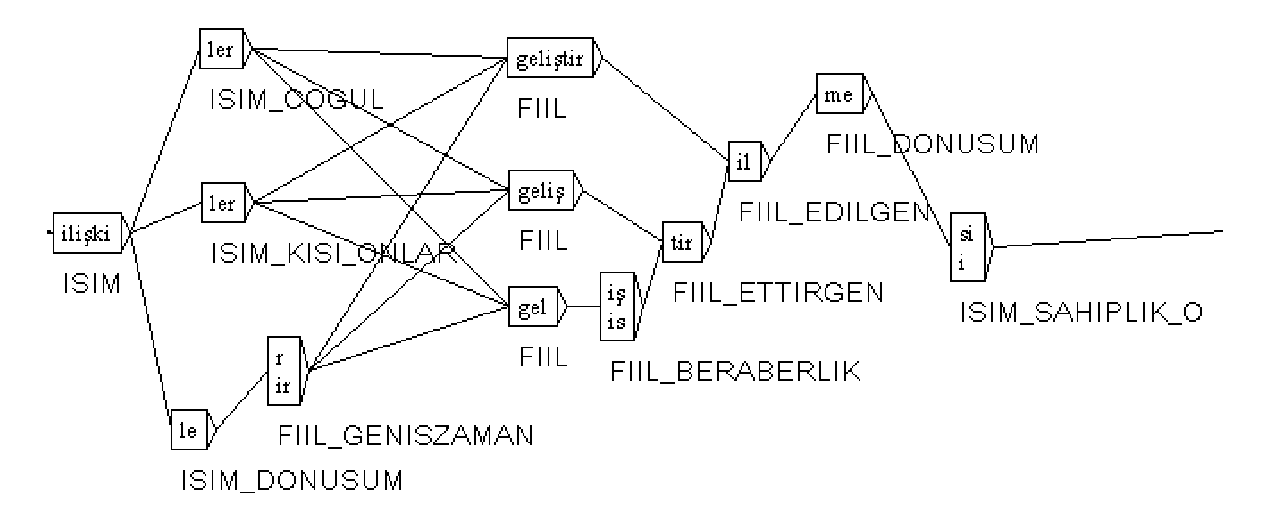


FIG. 4.1 – Portion d'automate du texte

De plus, l'automate du texte peut être utilisé pour appliquer au texte des grammaires ELAG [Lapo 98] de levée d'ambiguïtés. En effet, une application prometteuse de l'intégration du turc à Unitex est le développement de grammaires ELAG qui réduisent, grâce au contexte, les ambiguïtés les plus répandues. Par exemple, sachant que génitif et possessif apparaissent le plus souvent en

concomitance, la marque du possessif à la 2e personne du singulier pourrait être distinguée de celle de l'accusatif, lorsque le mot ambigu est précédé d'un mot au génitif.

En contrepartie, il n'est pour l'instant pas possible d'effectuer de recherches de motifs sur l'automate du texte, ce qui rend insuffisante cette forme de représentation du turc en Unitex.

L'automate du texte est normalement généré à travers les dictionnaires et les graphes de normalisation. Dans notre cas, il nous faudra transformer la sortie de l'analyseur choisi en un graphe de normalisation remplaçant chaque mot du texte par une suite d'entrées morphémiques (racines et suffixes).

4.2.4 Discussion

La solution A cause une perte remarquable d'information morphologique et ne permet pas d'exploiter pleinement la recherche de motifs dans Unitex. La solution B présente l'inconvénient de perdre l'ordre d'occurrence des suffixes, mais nous avons vu que ce problème peut être en partie réglé en convertissant un sous-ensemble des noms de suffixe en codes flexionnels. Enfin la solution C peut servir à visualiser les différentes segmentations possibles de chaque forme et à développer des grammaires de levée d'ambiguïté, mais elle est insuffisante car non compatible avec le moteur de recherche de motifs d'Unitex.

Nous allons donc implémenter les solutions B (avec conversion des noms des suffixes flexionnels en codes flexionnels) et C. Nous ferons ceci par la création d'un outil capable de convertir la sortie de Zemberek soit en un dictionnaire DELAF, soit en un graphe de normalisation qui sera utilisé pour produire l'automate du texte. Les détails techniques de cette réalisation sont présentés au chapitre suivant.

Chapitre 5

Réalisation

Les réalisations techniques de ce projet comprennent la conception d'un *package* Java gérant la représentation d'un texte turc au moyen des formalismes reconnus par Unitex, ainsi que quelques modifications apportées à l'analyseur de Zemberek.

5.1 Le package *toUnitex*

Le package `toUnitex` sert de liaison entre l'analyseur morphologique de la bibliothèque de traitement de texte Zemberek, et la plate-forme multilingue de traitement de corpus Unitex.

Dans Zemberek une forme qui a été analysée est associée à un tableau d'objets `Kelime` 'mot', un pour chaque analyse. Les attributs de la classe `Kelime` sont un objet de type `Kok` 'racine' et une liste d'objets de type `Ek` 'suffixe'. Ci-dessous sont reportées les trois analyses[1] de la forme *hakları* ('les droits (acc.)', 'ses droits', 'leur(s) droit(s)') produites par la classe de test `DemoMain`. Pour plus d'exemples se référer à l'Annexe A.

```
hakları
[ Kok: hak, ISIM ] Ekler: ISIM_COGUL_LER + ISIM_BELIRTME_I
[ Kok: hak, ISIM ] Ekler: ISIM_COGUL_LER + ISIM_SAHIPLIK_O_I
[ Kok: hak, ISIM ] Ekler: ISIM_SAHIPLIK_ONLAR_LERI
```

Les racines, `Kok`, sont des chaînes caractérisées par une catégorie : `KelimeTipi` *`tip`*, alors que les suffixes, `Ek`, sont caractérisés par un nom : `String` *`ad`*[2]. Selon la ressource Unitex que l'on

[1] *ISIM* 'nom', *ISIM_COGUL_LER* 'pluriel', *ISIM_BELIRTME_I* 'accusatif', *ISIM_SAHIPLIK_O_I* 'possessif 3e pers. sing.', *ISIM_SAHIPLIK_ONLAR_LERI* 'possessif 3e pers. plur.'. Cf. Annexe B.

[2] Par convention le premier suffixe de la liste est la catégorie de la racine concaténée à '_KOK', par ex. ISIM_KOK, FIIL_KOK etc.

souhaite générer, ces données doivent être formatées différemment.

Le package `toUnitex` contient les classes `Analyzer`, `ZemberekToUnitex` et `WordDelaEntries`. La classe `Analyzer` contient le programme principal qui gère arguments et options donnés en ligne de commande, et contrôle la chaîne d'opérations suivante :

– lecture du fichier contenant le texte turc,
– création du lexique des formes du texte,
– génération du dictionnaire DELAF *ou* de l'automate du texte, selon l'option choisie.

Ce qui suit est le message d'usage affiché par le programme principal de la classe `Analyzer` :

```
********
toUnitex.Analyzer <textToAnalyze> <in_encoding> <outFileName>
<out_encoding> <UnitexResourceToCreate> <correction>
********
<textToAnalyze> : filename of the text file to be analyzed
<in_encoding> : encoding of the text file to be analyzed
<outFileName> : filename where the Unitex resource has to be written
<out_encoding> : encoding of the resource file to be written
<UnitexResourceToCreate> :
 DELA_s : create DELA dictionary of the text with all suffixes
 coded as grammatical features (+SUFF1+SUFF2+...)
 DELA_i : create DELA dictionary of the text with inflectional
 suffixes coded as inflectional features (:xy...)
 FST : create morphological segmentation FST of the text
<correction> :
 YES : allow correction of Zemberek analysis
 NO : leave Zemberek analysis unchanged
```

5.1.1 Génération du dictionnaire DELAF

Disposer du dictionnaire DELAF des formes contenues dans le texte de travail nous permet d'y appliquer les fonctionnalités principales d'Unitex, parmi lesquelles la recherche de motifs et la génération de concordances qui servent à visualiser en contexte les résultats des recherches. L'implémentation présentée ci-dessous correspond à la solution B (cf. Sous-section 4.2.2).

Le DELAF du texte est généré par la méthode *`createDELAdictionary`* de la classe `ZemberekToUnitex`, qui, pour chaque analyse d'une forme, fait appel à l'une des deux fonctions de création d'entrée : *`DelaEntrySimple`* si l'option choisie est `DELA_s` (dictionnaire simple), *`DelaEntry`* si l'option choisie est `DELA_i` (dictionnaire *inflex*). Dans le premier cas la

forme est simplement concaténée à la racine suivie des noms des suffixes séparés par un plus (+). Par exemple, pour le mot *kitap-lar-ım-da* ‘dans mes livres’, on obtient[3] :

```
>kitaplarımda,kitap.ISIM_KOK+ISIM_COGUL_LER+ISIM_SAHIPLIK_BEN_IM
+ISIM_KALMA_DE
```

Dans le second cas, par contre, un algorithme de conversion des noms de suffixe en codes flexionnels est appliqué à l’analyse de la forme.

Génération des codes flexionnels

Comme il a été pointé à la Section 1.2, la suffixation en turc recouvre une quantité de phénomènes morphologiques divers. Il peut donc se révéler difficile d’accéder à l’information morphologique d’une forme si tous les suffixes sont reportés dans le même champ de l’entrée. Ceci est d’autant plus vrai que les traits du champ sémantique-grammatical ne sont pas ordonnés.

Pour générer le dictionnaire *inflex* nous allons définir une table de conversion des noms de suffixes en codes flexionnels, qui sera stockée dans le fichier *inflectionalCodes.txt* contenu dans le répertoire *kaynaklar/toUnitex*. En l’absence d’une nomenclature standard des suffixes turcs, le choix des codes est tout à fait arbitraire. Nous avons pourtant essayé d’assigner les codes de la façon la plus logique possible, étant données les contraintes strictes du formalisme DELAF, soit un seul caractère par code. C’est pourquoi les codes choisis font référence tantôt au nom turc du suffixe (suffixes de cas), tantôt au nom anglais (pluriel et suffixes verbaux). Afin d’améliorer la lisibilité du dictionnaire, les traits de flexion nominale reçoivent un code minuscule, ceux de flexion verbale un code majuscule. Les possessifs et les marqueurs de personne sont codés par un chiffre de un à six[4]. Voici donc des extraits de la table de conversion, disponible dans sa totalité à l’Annexe C :

```
# Plural -ler
ISIM_COGUL_LER      p       # pluriel (nominal)
# Possessive
ISIM_SAHIPLIK_BEN_IM     1       # mon/ma/mes
ISIM_SAHIPLIK_SEN_IN     2       # ton/ta/tes
ISIM_SAHIPLIK_O_I        3       # son/sa/ses
ISIM_SAHIPLIK_BIZ_IMIZ   4       # nos
ISIM_SAHIPLIK_SIZ_INIZ   5       # vos
```

[3] *ISIM_SAHIPLIK_BEN_IM* ‘possessif 1re pers. sing.”, *ISIM_KALMA_DE* ‘locatif’. Cf. Annexe B.

[4] Le même chiffre doit donc être lu comme un possessif s’il est associé à une base nominale, et comme un marqueur d’accord verbal s’il est associé à une base verbale.

```
ISIM_SAHIPLIK_ONLAR_LERI 6       # leur
# Case
ISIM_YONELME_E   y      # datif
ISIM_KALMA_DE    k      # locatif
ISIM_CIKMA_DEN   c      # ablatif
ISIM_BELIRTME_I  b      # accusatif
ISIM_TAMLAMA_IN  t      # génitif
[...]
# Negation
FIIL_OLUMSUZLUK_ME   N     # négation verbale
```

Selon cette table de conversion le mot *kitap-lar-ım-da* ‘dans mes livres’ donnera lieu à l’entrée suivante :

```
>kitaplarımda,kitap.ISIM_KOK:p1k
```

Au point de vue formel, la table de conversion est une application A d’un sous-ensemble des noms de suffixes flexionnels dans l’ensemble des caractères alphanumériques[5]. En effet l’ensemble de définition de A ne coïncide pas tout à fait avec la classification de notre grammaire de référence [Gök 05] car il exclut le suffixe pronominal *-ki*, ainsi que les suffixes subordonnants, les voix verbales (causative, passive, réflexive, réciproque) et les marqueurs de modalité verbale (-*(y)Abil, -(y)Iver,-(y)Agel,-(y)Ayaz,-(y)Akal,-(y)Adur*). Il nous paraît difficile de considérer ces suffixes comme dérivationnels d’un point de vue sémantique. De plus, les suffixes subordonnants impliquent le changement de la catégorie grammaticale du mot et peuvent donc entraîner la cooccurrence, dans la même forme, de traits de flexion verbale avec d’autres de flexion nominale, qui ne doivent pas, à notre avis, être mélangés dans le champ des traits flexionnels d’une même entrée.

Dans notre implémentation, les suffixes convertis en codes flexionnels doivent former une suite continue située à la fin de la forme. Autrement dit, aucun suffixe ne peut être converti s’il est suivi d’un suffixe non convertible. Poser cette condition revient à « protéger » les suffixes flexionnels précédant le suffixe pronominal *-ki* et ceux précédant les suffixes subordonnants. Par exemple une forme comme *kitap-lar-ım-da-ki* ‘celui qui est dans mes livres’ ne doit pas être marquée comme plurielle, même si elle contient le suffixe du pluriel ISIM_COGUL_LER :

```
>kitaplarımdaki,kitap.ISIM_KOK+ISIM_COGUL_LER+ISIM_SAHIPLIK_BEN_IM
+ISIM_KALMA_DE+ISIM_BULUNMA_KI
```

et non :

[5] L’application A n’est pas bijective, puisque le même code peut correspondre à plusieurs suffixes.

```
>kitaplarımdaki,kitap.ISIM_KOK+ISIM_BULUNMA_KI:p1k
```

Semblablement, dans le mot *git-me-diğ-im-e* litt. ‘au fait que je ne sois pas allé’, on traitera de flexion uniquement les suffixes de flexion nominale (*-im-e*) qui suivent le suffixe subordonnant (*-diğ-*), le suffixe négatif (*-me-*) faisant partie d’une autre construction morphologique, celle-ci de type verbale : *git-* ‘aller’ -> *git**me**-* ‘ne pas aller’. L’entrée correspondante sera donc :

```
>gitmediğime,git.FIIL_KOK+FIIL_OLUMSUZLUK_ME+FIIL_BELIRTME_DIK:1y
```

et non :

```
>gitmediğime,git.FIIL_KOK+FIIL_BELIRTME_DIK:N1y
```

L’algorithme implémenté par la méthode *DelaEntry* (cf. Annexe F) parcourt donc la séquence de suffixes de droite à gauche et génère un ensemble de codes flexionnels au travers de la table de conversion (*inflectionalCodes.txt*) fournie avec le package et modifiable par l’utilisateur. Les suffixes qui ont été trouvés dans la table de conversion sont retirés, en même temps, de la séquence d’informations sémantiques-grammaticales.

Factorisation d’entrées similaires

Nous avons noté auparavant que les mots turcs peuvent souvent admettre plus d’une analyse (cf. Sous-section 4.2.3). Le formalisme DELAF permet de réduire considérablement la taille du dictionnaire par la factorisation des entrées similaires, c’est-à-dire des entrées qui ne diffèrent que par leurs codes flexionnels (cf. Sous-section 3.1.2). Ce procédé peut être observé dans le DELAF du français :

```
>accepte,accepter.V+z1:P1s:P3s:S1s:S3s:Y2s
```

La méthode de génération d’entrées *DelaEntry* gère la factorisation d’entrées similaires à travers la classe `WordDelaEntries` (cf. Annexe F), dont une instance représente l’ensemble d’entrées DELAF associées à une forme. Ainsi l’entrée suivante :

```
>hürriyetleri,hürriyet.ISIM_KOK:pb:p3:6
```

remplace les trois entrées similaires :

```
>hürriyetleri,hürriyet.ISIM_KOK:pb
>hürriyetleri,hürriyet.ISIM_KOK:p3
>hürriyetleri,hürriyet.ISIM_KOK:6
```

Bien évidemment, aucune factorisation ne peut être appliquée aux entrées générées avec l'option `DELA_s`, du fait que le champ des codes flexionnels y est toujours vide.

Les Tables 5.1 et 5.2[6] montrent les différentes entrées générées selon l'option de dictionnaire choisie, pour deux formes, une nominale et une verbale.

Forme	**hürriyetleri**
Trad.	'les libertés(acc.)', 'ses libertés', 'leur(s) liberté(s)'
DELA_s	>hürriyetleri,hürriyet.ISIM_KOK+ISIM_COGUL_LER+ISIM_BELIRTME_I >hürriyetleri,hürriyet.ISIM_KOK+ISIM_COGUL_LER+ISIM_SAHIPLIK_O_I >hürriyetleri,hürriyet.ISIM_KOK+ISIM_SAHIPLIK_ONLAR_LERI
DELA_i	>hürriyetleri,hürriyet.ISIM_KOK:pb:p3:6

Table 5.1: Entrées simples et *inflex* associées à la forme nominale *hürriyetleri*

Forme	**gitmeliyim**
Trad.	(litt.) 'je suis avec le fait d'aller', 'je dois aller'
DELA_s	>gitmeliyim,git.FIIL_KOK+FIIL_DONUSUM_ME+ISIM_BULUNMA_LI +ISIM_KISI_BEN_IM >gitmeliyim,git.FIIL_KOK+FIIL_ZORUNLULUK_MELI +ISIM_KISI_BEN_IM
DELA_i	>gitmeliyim,et.FIIL_KOK+FIIL_DONUSUM_ME+ISIM_BULUNMA_LI:1 >gitmeliyim,et.FIIL_KOK:O1

Table 5.2: Entrées simples et *inflex* associées à la forme verbale *gitmeliyim*

5.1.2 Génération de l'automate du texte

Nous allons maintenant présenter l'implémentation de la solution C (cf. Sous-section 4.2.3).

L'automate du texte est construit dans Unitex à travers l'application des dictionnaires et/ou des graphes de normalisation. Les dictionnaires peuvent donner lieu à des ambiguïtés liées aux mots

[6] *FIIL_DONUSUM_ME* 'nominalisation verbale de type *-mA*', *ISIM_BULUNMA_LI* 'suff. dérivationnel (possession, origine)', *ISIM_KISI_BEN_IM* 'accord verbal 1re personne du singulier', *FIIL_ZORUNLULUK_MELI* 'mode verbal obligatif'. Cf. Annexe B.

composés. En français, par exemple, l'entrée du mot composé « carte postale » est en concurrence avec les deux entrées simples « carte » et « postale », ce qui va engendrer deux chemins parallèles dans l'automate du texte.

Au contraire, les graphes de normalisation sont utilisés soit pour remplacer les tokens par des formes normalisées, soit pour les segmenter en des unités plus petites. En français on peut citer le cas de l'article tronqué « l' » normalisable de quatre façons différentes (« le » déterminant, « le » pronom, « la » déterminant et « la » pronom). Plus intéressant est le cas du thaï, où les graphes de normalisation sont employés pour segmenter en mots le texte naturellement dépourvu de séparateurs graphiques.

Dans le cas du turc, nous allons remplacer chaque forme par la suite des morphèmes, racine plus suffixes, qui la composent. Lorsqu'on donne à la classe `Analyzer` l'option `FST`, la méthode *`createSegmentationFST`* de la classe `ZemberekToUnitex` va générer le graphe de normalisation correspondant à la segmentation des formes du texte lu. Cette méthode fait appel, pour chaque analyse d'une forme, à la fonction de création d'état de graphe *`segmentationFSTstate`*. Un état est composé d'un symbole d'entrée et d'un symbole de sortie séparés par le caractère « / » (par référence à la notion de transducteur, cf. Sous-section 3.1.2). Le symbole d'entrée est la forme rencontrée dans le texte, alors que le symbole de sortie est une suite d'entrées dictionnairiques au format DELAF. La production de ce genre de graphe nous dispense donc de générer le dictionnaire des formes du texte.

Nous donnons ici, en guise d'exemple, les états du graphe de normalisation[7] associés aux formes *ilişkiler* 'les relations' et *geliştirilmesi* litt. 'son développement' dont l'automate a été présenté à la Sous-section 4.2.3 :

```
%ilişkiler/{ilişki,.ISIM} {ler,.ISIM_COGUL}
%ilişkiler/{ilişki,.ISIM} {ler,.ISIM_KISI_ONLAR}
%ilişkiler/{ilişki,.ISIM} {le,.ISIM_DONUSUM} {r,ir.FIIL_GENISZAMAN}
%geliştirilmesi/{geliştir,.FIIL} {il,.FIIL_EDILGEN} {me,.FIIL_DONUSUM}
 {si,i.ISIM_SAHIPLIK_O}
%geliştirilmesi/{geliş,.FIIL} {tir,.FIIL_ETTIRGEN} {il,.FIIL_EDILGEN}
 {me,.FIIL_DONUSUM} {si,i.ISIM_SAHIPLIK_O}
%geliştirilmesi/{gel,.FIIL} {iş,is.FIIL_BERABERLIK} {tir,.FIIL_ETTIRGEN}
 {il,.FIIL_EDILGEN} {me,.FIIL_DONUSUM} {si,i.ISIM_SAHIPLIK_O}
```

[7] *ISIM_KISI_ONLAR* 'accord verbal 3e personne du pluriel', *ISIM_DONUSUM* 'nominalisation verbale', *FIIL_GENISZAMAN* 'mode verbal aoriste', *FIIL_EDILGEN* 'voix verbale passive', *FIIL_ETTIRGEN* 'voix verbale causative', *FIIL_BERABERLIK* 'voix verbale réciproque'. Cf. Annexe B.

Nous pouvons observer que chaque morphème ne reçoit qu'un trait sémantique-grammatical : ce trait indique soit la catégorie des racines, soit le nom des suffixes. Dans ce modèle de représentation l'information morphologique est totalement mise à plat, par opposition au modèle basé sur dictionnaire. De cette façon, aucune contrainte ne s'impose à l'expression de l'information produite par l'analyseur, mais l'accès et le traitement de cette information par les fonctionnalités d'Unitex n'en sont pas pour autant garantis. L'intérêt de cette solution, nous le rappelons, réside dans la possibilité de visualiser les ambiguïtés morphologiques au moyen de l'automate du texte et d'y appliquer des grammaires de désambiguïsation par le contexte.

5.2 Modifications apportées à Zemberek

Les modifications apportées à Zemberek se situent à divers niveaux.

Une classe native de la bibliothèque, `net.zemberek.araclar.turkce.YaziIsleyici`, a été directement touchée par :

– l'ajout de la méthode *yaziOkuyucuKodSecimi*, équivalente à la méthode native *yaziOkuyucu*, mais avec un argument de plus pour l'encodage du texte analysé (fixé à ISO-8859-9 dans *yaziOkuyucu*) ;
– l'ajout de la méthode *metninSozlugu* qui segmente le texte lu et crée le lexique des formes.

Les autres changements concernent la base de connaissances linguistiques incluse dans la bibliothèque, située dans le repertoire *kaynaklar/tr*. Les tests effectués montrent, en effet, que plusieurs conjonctions et postpositions sont analysés comme noms (*ISIM*), au lieu de recevoir la bonne étiquette (*BAGLAC* pour les conjonctions, *EDAT* pour les postpositions). Ces informations se trouvent dans le dictionnaire des racines, disponible uniquement sous forme de fichier binaire. Dès lors nous n'avons pas d'autre choix que de corriger les analyses à posteriori, c'est-à-dire après qu'elles aient été produites par l'analyseur de Zemberek et avant de les formater en dictionnaire ou en graphe de normalisation. Ce procédé de post-édition, optionnel, est réalisé par la méthode *correctAnalysis* du package `toUnitex` lorsqu'on donne la valeur `YES` à l'option `<correction>`. La table de correction des catégories appliquée par la méthode *correctAnalysis* est stockée dans le fichier *rootCategories.txt* du répertoire *kaynaklar/toUnitex*.

Le lexique des suffixes nécessitait aussi, à notre avis, quelques modifications qui ont pu être apportées directement au fichier *ek_tr.xml*. Les rectifications concernent trois suffixes dont les dé-

finitions violent le paradigme de la flexion nominale donné à la Table 1.2 :

1. le suffixe ISIM_TAMLAMA_I a été supprimé et sa morphotactique a été incluse dans le suffixe du possessif (SAHIPLIK_O_I)[8]. L'analyse suivante de la forme *evinde* ('dans sa maison', 'dans ta maison') est donc éliminée :

   ```
   [ Kok:  ev, ISIM ] Ekler:  ISIM_TAMLAMA_I + ISIM_KALMA_DE
   ```

2. le suffixe du génitif ISIM_TAMLAMA_IN (*-(n)In*) est un suffixe de cas. Il ne peut pas, de ce fait, être suivi d'un autre suffixe de cas. Cette autre analyse de la forme *evinde* est aussi invalidée, puisque le suffixe du locatif ISIM_KALMA_DE est un suffixe de cas :

   ```
   [ Kok:  ev, ISIM ] Ekler:  ISIM_TAMLAMA_IN + ISIM_KALMA_DE
   ```

   En résumant, voici le résultat desdits changements sur les analyses de la forme *evinde* :

   evinde
   [Kok: evin, ISIM] Ekler: ISIM_KALMA_DE
   ~~[Kok: ev, ISIM] Ekler: ISIM_TAMLAMA_IN + ISIM_KALMA_DE~~
   ~~[Kok: ev, ISIM] Ekler: ISIM_TAMLAMA_I + ISIM_KALMA_DE~~
   [Kok: ev, ISIM] Ekler: ISIM_SAHIPLIK_SEN_IN + ISIM_KALMA_DE
   [Kok: ev, ISIM] Ekler: ISIM_SAHIPLIK_O_I + ISIM_KALMA_DE

3. le suffixe ISIM_TANIMLAMA_DIR (*-dIr*) a valeur de copule et ne fait pas partie du groupe des suffixes de cas, comme démontré par cet exemple, où il est à son tour précédé d'un suffixe de cas génitif :

   Hocamındır. 'C'est à mon professeur.'
   hoca(professeur) +*m*[Poss1sing] +*ın*[Gén] +*dır*[Copule]

   Grâce à cette modification, la forme *hocamındır*, qui n'était pas reconnue par la version originelle de Zemberek, est désormais analysée comme suit :

   ```
   hocamındır
   [ Kok: hoca, ISIM ] Ekler: ISIM_SAHIPLIK_BEN_IM + ISIM_TAMLAMA_IN
   + ISIM_TANIMLAMA_DIR
   ```

[8] Le suffixe *-(s)I* est étiqueté par Zemberek de deux façons :
– comme suffixe possessif à la 3e personne du singulier (ISIM_SAHIPLIK_O_I), par ex. *Ayşe'nin çanta**sı*** 'le sac d'Ayşe',
– comme marque de tête de mot composé (ISIM_TAMLAMA_I), par ex. *tarla kuş**u*** litt. 'oiseau des champs', 'alouette'.
Cette distinction concerne la fonction sémantique du suffixe, mais ne justifie pas le redoublement des analyses morphologiques, d'autant plus que la distinction n'introduit aucune différence morphotactique. Les grammaires consultées [Gök 05], [Bazi 87] ont confirmé notre avis.

Chapitre 6

Évaluation

Nous avons évalué le travail réalisé sur trois corpus en langue turque, différents en taille et en genre :

- *Fables* : un choix de 12 fables de La Fontaine téléchargées sur un site de contes et récits.[1] (≃3000 mots),
- *UDHR* : la Déclaration Universelle des Droits de l'Homme (≃3000 mots),
- *RevOTAN* : un recueil de 35 articles de politique internationale publiés dans la Revue de l'OTAN[2] entre 2005 et 2006 (≃130000 mots).

Les sections qui suivent présentent les résultats obtenus par la représentation de ces trois corpus en Unitex à travers les techniques décrites au Chapitre 5.

6.1 Dictionnaires : données numériques

Le tableau 6.1 montre quelques données numériques obtenues en générant les dictionnaires DELAF (simple et *inflex*) des trois corpus. La colonne *Texte* donne le nombre d'occurrences et de formes du corpus. La colonne *Formes inconnues* donne la quantité, en nombre et en pourcentage, des formes qui n'ont pas été reconnues par l'analyseur de Zemberek. La colonne *DELA_s* contient la taille, en nombre d'entrées et de kilo-octets (KB), du dictionnaire simple des formes du corpus, généré avec le package `toUnitex`. Les mêmes données concernant le dictionnaire *inflex* sont reportées à la colonne *DELA_i*.

[1] `www.sevdamisali.net`. Aucun contrôle orthographique n'a été effectué sur ce corpus.
[2] `http://www.nato.int/docu/review.htm`

	Texte		Formes inconnues		DELA_s		DELA_i	
Corpus	occurr.	formes	nb	%.	entrées	KB	entrées	KB
Fables	2977	796	29	3,6%	1426	136	1207	84
UDHR	3160	720	17	2,4%	1322	144	1065	84
RevOTAN	129592	12970	908	7,0%	25172	3200	18919	1700

TAB. 6.1 – Évaluation des dictionnaires sur les trois corpus

Formes inconnues. Le taux des formes inconnues est important surtout dans les corpus *RevOTAN* (7%), moins dans les autres. Les formes inconnues du corpus *RevOTAN* incluent des centaines de noms propres d'organisations, de lieux et de personnes (*PKK, Oklahoma, Milosevic, ...*), ainsi que des mots techniques et des néologismes (*modernizm, paralize, radyolojik, ...*)[3]. L'oubli, dans le dictionnaire des racines, du seul mot *hükümet* 'gouvernement' est à l'origine de 16 formes fléchies inconnues. Enfin la marque du plus-que-parfait *-mIş-DI* n'est pas reconnue comme une suite de suffixes valide par Zemberek, ceci étant aussi probablement un oubli parmi les règles morphotactiques.

Les formes inconnues du corpus *UDHR* (2,4%) comprennent des termes légaux et d'autres termes rares de la langue (*cürüme* 'corruption', *ayırdedici* 'discriminant', *(hiçbir) veçhile* 'd'aucune façon'), des graphies alternatives à celles recensées par le dictionnaire de Zemberek (*mevzubahis - mevzuubahis* 'en question'), des fautes d'orthographe (**malümat - malumat* 'information', **mualele - muamele* 'traitement').

Enfin, dans le corpus *Fables*, le taux de formes inconnues (3,6%) s'explique par l'habitude des internautes turcs de noter les caractères spéciaux par leurs correspondants sans diacritiques. Ainsi dans le même site codé en Unicode il est fréquent de trouver des graphies correctes alternées avec des graphies simplifiées, telle **cikmak* au lieu de *çıkmak*, ces dernières n'étant naturellement pas incluses dans le dictionnaire des racines de Zemberek[4].

Taux d'ambiguïté morphologique. Le nombre moyen d'analyses par forme nous renseigne sur le taux d'ambiguïté morphologique d'une langue. Ce nombre, calculé en divisant le nombre d'en-

[3]La question des néologismes et des noms propres concerne tous les analyseurs qui se basent sur des algorithmes guidés par la racine. Une solution réside dans les algorithmes de type *affix stripping*, présentés à la Section 2.3.

[4]Ce problème pourrait être réglé en se servant de la fonction de correction orthographique de la bibliothèque Zemberek.

trées DELA_s par le nombre de formes du corpus[5], est égal à 2,08 pour le corpus *Fables*, 1,83 pour le corpus *UDHR* et 1,46 pour le corpus *RevOTAN*. La taille du corpus paraît influer sur cette donnée, plutôt que le genre textuel.

Réduction de la taille du dictionnaire. La conversion des noms de suffixe en codes flexionnels, unie à la factorisation des entrées similaires, permet une réduction sensible de la taille des dictionnaires générés. Le dictionnaire *inflex* du corpus *Fables* a 15,3% d'entrées en moins que le dictionnaire simple et occupe 38,2% d'espace mémoire en moins. Semblablement, pour les corpus *UDHR* et *RevOTAN* la diminution du nombre d'entrées est égale à 19,4% et 24,8% respectivement, la réduction de la taille du fichier est égale à 41,7% et 46,8% respectivement. On observe donc que la réduction du dictionnaire est d'autant plus importante que la taille du corpus augmente.

6.2 Recherche de motifs

Nous allons maintenant montrer l'utilité du travail réalisé par des exemples de recherche de motifs, s'appuyant sur la syntaxe du dictionnaire *inflex*.

Les motifs qui font appel aux informations contenues dans les dictionnaires du texte sont dits, dans Unitex, masques lexicaux. Ils se distinguent des simples unités lexicales, parce qu'ils sont notés entre chevrons. Grâce au DELAF généré, nous pouvons désormais utiliser les masques lexicaux d'Unitex sur le turc.

L'intérêt premier d'analyser morphologiquement un texte est de pouvoir travailler avec les formes canoniques des mots qui le composent. Voici des portions des concordances produites à partir du corpus *Fables* pour le substantif *avcı* 'chasseur' et pour le verbe *de-* 'dire' en lançant les requêtes <avcı> et <de> :

, avcının cıplak topuğundan ısırmış.{S} **Avcı** da acı ile haykırmış.{S} Avcının sesini
a cıkmış.{S} Bu sırada oradan gecen bir **avcı**, güvercini görünce sevinmiş.{S} Ne güzel
ış {S}Aradan zaman geçmiş, Aslan birgün **avcıların** kurduğu tuzağa yakalanmış. {S}Aslan
p nişan almış. {S}Karınca bunu görunce, **avcının** cıplak topuğundan ısırmış.{S} Avcı da
ırmış.{S} Avcı da acı ile haykırmış.{S} **Avcının** sesini duyan güvercinde havalanıp ucm

[5] Nous rappelons que le dictionnaire simple DELA_s a autant d'entrées qu'il y a d'analyses produites par l'analyseur de Zemberek. Ceci n'est pas vrai pour le dictionnaire *inflex* du fait de la factorisation des entrées similaires.

ene aldırmamış. " Siz rahatınıza bakın" **demiş**.{S} Yemlerini yiyip uyumuşlar. . {S}Ert
yordun.{S} Bak. senin canını kurtardım, **demiş**. {S}Aslan, böylece yapılan bir iyiliğin
olur benimda sana bir iyiliğim dokunur, **demiş**. {S}Aslan farenin bu sözlerine gülerek :
ne takıp terketmiş yuvasını. {S}Hani ne **derler** insanın dostu da kendisidir, düşmanı d
p, yavrularımızı yemek. " {S}Kedi böyle **deyince** kartalı bir korkudur almış.{S} Ne yap
ir faresin, bana ne iyiliğin dokunur ki **deyip**, fareye acımış ve fareyi bırakmış. {S}F
görünce sevinmiş.{S} Ne güzel bir av. " **diye** düşünmüş.{S} Her şeyden habersiz güverci
: " Hayrola, ne var ?{S} Ne oluyor ? " **diye** sormuş. {S}Kedi : " Daha ne olsun " diye
ptığı gibi yılanı üç parçaya bölmüş. {S} **Diyeceğimiz** şu ki ;{S} acımak, iyilik güzel ş

Les masques lexicaux permettent, en outre, de préciser les traits sémantiques-grammaticaux et les codes flexionnels de la forme recherchée. La requête <avcı.ISIM_KOK :t>[6] rapporte les formes de *avcı* au génitif (***t** tamlama*), <de.FIIL_KOK :E> les formes de *de-* au passé de non-constatation (***E** evidential past*) :

ış {S}Aradan zaman geçmiş, Aslan birgün **avcıların** kurduğu tuzağa yakalanmış. {S}Aslan
p nişan almış. {S}Karınca bunu görunce, **avcının** cıplak topuğundan ısırmış.{S} Avcı da
ırmış.{S} Avcı da acı ile haykırmış.{S} **Avcının** sesini duyan güvercinde havalanıp ucm

ene aldırmamış. " Siz rahatınıza bakın" **demiş**.{S} Yemlerini yiyip uyumuşlar. . {S}Ert
yordun.{S} Bak. senin canını kurtardım, **demiş**. {S}Aslan, böylece yapılan bir iyiliğin
olur benimda sana bir iyiliğim dokunur, **demiş**. {S}Aslan farenin bu sözlerine gülerek :

Des contraintes plus complexes peuvent être exprimées sur le champ des codes flexionnels :

- conjonction de codes flexionnels. Par ex. <avcı.ISIM_KOK :pt> pour les formes de *avcı* au pluriel génitif :

ış {S}Aradan zaman geçmiş, Aslan birgün **avcıların** kurduğu tuzağa yakalanmış. {S}Aslan

- disjonction de codes flexionnels. Par ex. <aslan.ISIM_KOK :t :y> pour les formes de *aslan* ‘lion’ au génitif ou au datif (***y** yönelme*) :

.{S} Aslanı tuzaktan kurtarmış. {S}Fare **aslana** : - Beni küçük diye beğenmiyordun.{S} B
mış.{S} Oradan geçmekte olan minik fare **aslanın** bu durumunu görmüş.{S} Hemen dişleri
iş, yatmış uyuyormuş.{S} Minik bir fare **aslanın** üzerinde dolaşmaya başlamış.{S} Aslan

Il est également possible d’omettre le lemme recherché. Ainsi la requête <FIIL_KOK :NAZ> rapporte tous les verbes au négatif, mode aoriste[7] conditionnel (***N** négation, **A** aoriste, **Z** copule conditionnelle*). Voici une partie des résultats obtenus dans le corpus *RevOTAN* :

[6] Les contraintes portant sur les codes flexionnels doivent obligatoirement être précédées par au moins un code sémantiques-grammatical. Dans ces exemples, nous employons le nom de suffixe qui indique, par convention, la catégorie de la racine.

[7] « L’aoriste exprime l’action dans sa généralité, sans actualisation ni limitation de durée. » [Bazi 87].

la bahsetmişti.{S} Bu sorunları ciddiye **almazsak** batı uygarlığı ve bizzat varoluşumuz
in kendini yenilemeye çalışan NATO bunu **başaramazsa** birlikte çalışabilirlik konusunda
S} ülkeler iki yıllık kuvvet tahsisinde **bulunamazlarsa**, belirli bir kriz mukabele ope
kikada taahhüt koparmak için kahramanca **çabalamazsa**, CJSOR'un tamamlanamaması tehdidi
yında NATO'nun – son dakikada bir engel **çıkmazsa** – NATO Mukabele Kuvveti'nin (NMK)
İttifak eğer Afganistan sorunu ile baş **edemezse** NATO'nun varoluş nedeni de anlamını

Les codes du champ sémantique-grammatical peuvent être exclus par le caractère moins (-). Le motif <FIIL_KOK+FIIL_YETENEK_EBIL-FIIL_EDILGEN_IL> se lit donc « verbes en modalité *possible* à la voix active » :

– le suffixe FIIL_YETENEK_EBIL rajoute au verbe le sens de possibilité, capacité (par ex. *git-* 'venir' -> *gid**ebil**-* 'pouvoir venir') ;
– le suffixe FIIL_EDILGEN_IL est passivant. En le niant, nous pouvons sélectionner les verbes à la voix active.

Ci-dessous une partie des résultats obtenus dans le même corpus :

da müttefikler arasında sürtüşmeye yol **açabilir**.{S} Bir kültürden gelen barışı korum
nel işlevleri konusunda NATO'dan yardım **alabilmesidir**.{S} BM'nin aksine Avrupa Birliğ
merkezi olarak yapılandırılan bu merkez **alınabilen** tüm bilgiyi toplayıp, bir araya ge
üş görünüyor ? {S}NATO'nun adaptasyonunu **anlayabilmek** için İttifak'ın bütünlüğünün ark
çalışan NATO bunu başaramazsa birlikte **çalışabilirlik** konusunda artan problemlerle k
n kullanımından doğacak sonuçlarla başa **çıkabilecek** bir birim tahsis etmeyi önermişti
li biçimde konuşlandırabilme yeteneğini **değerlendirebilmek** için daha iyi kullanılabil

Pour conclure nous donnons un exemple de motif impliquant la concaténation de plusieurs unités lexicales : le motif <ISIM_KOK :t> <ISIM_KOK :3> reconnaît les syntagmes nominaux constitués d'un substantif au génitif suivi de la tête au possessif à la 3e personne du singulier. Voici une partie des résultats obtenus dans le corpus *UDHR* :

ususunun, hürriyetin, adaletin ve dünya **barışının temeli** olmasına, {S}İnsan haklarını
ürriyetlerinin kullanılmasında, sadece, **başkalarının haklarının** ve hürriyetlerinin ge
S}Madde 13 {S}1.{S} Herkes herhangi bir **devletin sınırları** dahilinde serbestçe dolaşm
}Yaşamak, hürriyet ve kişi emniyeti her **ferdin hakkıdır**. {S}Madde 4 {S}Hiç kimse köle
şahsının haysiyet ve değerine, erkek ve **kadınların eşitliğine** olan imanlarını bir ker
ücrete hakkı vardır. {S}3. çalışan her **kimsenin kendisine** ve ailesine insanlık haysi
ve insan haklarıyla ana hürriyetlerine **saygının kuvvetlenmesini** hedef almalıdır.{S}

En utilisant les expressions régulières nous pouvons éteindre le motif pour arriver a repérer des syntagmes plus complexes, où la tête est séparée du complément par d'autres unités lexicales. Parmi les résultats de la requête <ISIM_KOK :t> <MOT>* <ISIM_KOK :3> lancée sur le même corpus on trouve les occurrences suivantes :

arının tanınması hususunun, hürriyetin, **adaletin ve dünya barışının** temeli olmasına,
rensel beyannamesi {S}Önsöz {S}İnsanlık **ailesinin bütün üyelerinde** bulunan haysiyetin
ususunun, hürriyetin, adaletin ve dünya **barışının temeli** olmasına, {S}İnsan haklarını
ürriyetlerinin kullanılmasında, sadece, **başkalarının haklarının** ve hürriyetlerinin ge
fade edilir. {S}Madde 22 {S}Her şahsın, **cemiyetin bir üyesi** olmak itibariyle, sosyal
e tam rızasıyla yapılır. {S}3.{S} Aile, **cemiyetin tabii ve temel unsurudur**, cemiyet v
isnadın tespitinde, tam bir eşitlikle, **davasının bağımsız ve tarafsız bir mahkeme tarafından**
S}Madde 13 {S}1.{S} Herkes herhangi bir **devletin sınırları** dahilinde serbestçe dolaşm
evlenme konusunda, evlilik süresince ve **evliliğin sona ermesinde eşit hakları** haizdir

Pour les autres options de la recherche de motifs d'Unitex, se référer au manuel d'utilisation [Paum 06].

6.3 Discussion

Nous avons démontré qu'un corpus turc peut désormais être chargé sur la plate-forme et traité en utilisant des ressources morphologiques de base. Les dictionnaires générés par le package implémenté sont lus et interprétés sans faute par Unitex. Les graphes de normalisations sont traités comme attendu par le programme de création de l'automate du texte. Enfin les exemples de requêtes donnés à la section précédente illustrent les avantages de représenter le turc au moyen du dictionnaire *inflex*.

Le modèle morphologique que nous avons choisi montre pourtant quelques limites.

En premier lieu il est impossible de contraindre l'ordre d'occurrence des suffixes codés dans le champ des informations sémantiques-grammaticales. Cependant la génération de codes flexionnels diminue notablement le nombre de ces suffixes, d'ailleurs nous n'avons pas encore rencontré de cas où ceci constituerait un véritable problème aux fins du traitement de corpus.

Deuxièmement la recherche de motifs d'Unitex permet la négation des codes sémantiques-grammaticaux, mais pas celle des codes flexionnels. Il n'est donc pas possible d'imposer que, par exemple, le substantif recherché ne soit pas en nombre pluriel. Ceci constitue un problème dans la mesure où, en turc comme en français, certains traits grammaticaux ne sont marqués par aucun suffixe particulier. Dès lors, on dira qu'ils correspondent à $\emptyset$-suffixe. C'est le cas du nombre singulier, du cas nominatif et de la voix passive, entre autres. Nous avons réussi à exprimer cette dernière, par la négation de la voix active, codée dans le champ sémantique-grammatical. Au contraire, le singulier et le nominatif ne sont nullement exprimables dans la recherche de motifs d'Unitex avec le dictionnaire que nous avons conçu. Il faudrait, à cet égard, assigner à tous les substantifs des traits par défaut (singulier, nominatif) sauf en présence de suffixes les contredisant (pluriel, suffixes de cas).

Pour le reste, nous estimons ne pas avoir introduit de pertes d'information en intégrant à Unitex l'analyse produite par Zemberek. Par contre nous soulignons encore une fois que le dictionnaire des racines et celui des suffixes employés par Zemberek nécessiteraient une bonne révision. Des oublis et des erreurs ont aussi été observés parmi les règles morphotactiques.

Étant données les ressources dont nous disposions et les contraintes imposées par les formalismes d'Unitex, les réalisations techniques mises en place au cours de ce projet sont tout à fait fonctionnelles et permettent de répondre aux besoins initiaux.

L'approche choisie est donc satisfaisante, bien qu'elle puisse être davantage affinée, par exemple en dotant le générateur de dictionnaire *inflex* d'un dispositif d'assignation de codes flexionnels par défaut afin de régler convenablement le problème des traits « $\emptyset$-suffixe ». Le chapitre suivant montre comment ce même problème peut être contourné dans l'écriture des graphes syntaxiques d'Unitex, en utilisant les « contextes » pour exprimer la négation d'un ensemble de traits [Paum 06, pp. 94-95].

Chapitre 7

Application

Une fois rodé le modèle morphologique, nous pouvons passer à l'application concrète des ressources générées au traitement automatique du turc. Le présent chapitre est consacré à l'exposition d'une grammaire de classification des propositions subordonnées non finies, entièrement développée à travers les graphes d'Unitex en utilisant la syntaxe du dictionnaire *inflex*.

Comme anticipé à la Section 1.2, la suffixation est le moyen privilégié de rendre la subordination en turc ([Gök 05], p. 90). Les « subordonnées non finies » (SNF) sont construites autour d'une forme verbale non finie. Elles peuvent inclure une postposition, mais pas de conjonctions. Les SNF s'opposent aux « subordonnées finies », qui, elles, reposent sur un verbe fini et font emploi d'une conjonction subordonnante d'origine étrangère. Les subordonnées finies, moins fréquentes que les SNF en turc, sont presque absentes de la langue écrite formelle, mais sont très employées à l'oral. Elles ne sont tout de même pas traitées par notre grammaire.

L'idée de classifier et étiqueter les formes verbales non finies régissant des propositions subordonnées est née de la difficulté qu'éprouvent les apprenants du turc dans la compréhension et la traduction de ces éléments complexes de la langue. Cette difficulté, naturellement due à la distance structurelle entre la subordination turque et celle des langues indo-européennes, parmi d'autres, a été largement étudiée (par exemple par [Aydi 07]). Ce qui est plus surprenant, il a été démontré qu'un type particulier de SNF, à savoir les propositions relatives, pose des problèmes d'apprentissage aux enfants turcs mêmes [Cagr 05].

Pour revenir au domaine du traitement automatique de la langue, les SNF représentent un défi évident pour les programmes d'alignement de corpus bilingue, et encore plus pour les systèmes de traduction automatique ou assistée par ordinateur.

L'expérience d'étiquetage des SNF que nous allons présenter doit être vue comme première

étape fondamentale d'un projet de plus long haleine concernant la classification automatique des SNF, et dont l'objectif ultime serait l'analyse automatique de la proposition relative.

En développant une grammaire pour l'étiquetage des SNF, nous essayons d'exploiter l'information morphologique contenue dans le mot et dans son contexte immédiat, afin d'enrichir le texte avec des informations syntaxiques et sémantiques. Aux fins de notre exposé, cela représente une ultérieure illustration des possibilités offertes à la recherche par la ressource mise au point.

7.1 Les suffixes subordonnants

Les formes verbales non finies régissant des SNF se reconnaissent à la présence d'un suffixe subordonnant (SS). Les suffixes subordonnants sont : *-DIK, -(y)AcAK, -mA, -mAK, -(y)An, -(y)Iş*, plus une douzaine de suffixes formant des converbes[1]. Ils s'appliquent à des racines verbales pour les nominaliser. De ce fait ils ont un statut particulier : tout en étant classés comme flexionnels, ils impliquent le changement de catégorie du radical, qui est normalement le signe de la dérivation.

Les formes verbales non finies sont divisées en trois groupes par [Gök 05, p. 90] :

– les **noms verbaux** régissent des **propositions substantives** ou complétives (PS),
– les **participes** régissent des **propositions relatives** (PR),
– les **converbes** régissent des **propositions adverbiales** ou circonstancielles[2] (PA).

Alors que la plupart des SS turcs forment seulement un de ces trois types de verbe non fini, il en existe quelques uns qui peuvent rentrer dans plusieurs catégories. La difficulté majeure de la classification automatique des SNF réside donc dans la désambiguïsation des SS appartenant à ce groupe très productif (*-DIK, -(y)AcAK, -mA et -mAK*). La Table 7.1 illustre par des exemples les trois fonctions jouées par le suffixe *-DIK*, qui sont difficilement distinguables par le contexte immédiat de la forme verbale où le suffixe apparait.

Exception faite pour trois converbes (*-DIkça*, *-DIktAn* et *-AcAk kadar*), les suffixes *-DIK* et *-(y)AcAK*[3] sont toujours suivis d'un marqueur de personne, plus précisément d'un suffixe possessif

[1] 'Converbe', de l'allemand *Konverb*, est le terme forgé pour désigner les formes verbales non finies exprimant une subordination de type circonstanciel ou adverbial dans la langue mongole. Le terme a depuis été étendu à la description des langues turciques et ouraliennes.

[2] Nous préférons les termes 'substantive' à 'complétive, et 'adverbiale' à 'circonstancielle', car plus proches des termes anglais employés par [Gök 05].

[3] Ces deux suffixes subordonnants très productifs sont corrélés : ils apparaissent souvent dans les mêmes contextes, mais avec des valeurs temporelles différentes. En effet *-DIK* réfère à une action passée ou présente, tandis que *-(y)AcAK*

NC :	*oku-**duğ**-um*	->	*Kitabı oku**duğ**um-u biliyor.*
	'le fait que j'aie lu'		'Il sait que j'ai lu le livre.'
RC :	*oku-**duğ**-um kitap*	->	*Oku**duğ**um kitap nerede ?*
	'le livre que j'ai lu'		'Où est le livre que j'ai lu ?'
AC :	*oku-**duğ**-um-da*	->	*Kitabı oku**duğ**umda çocuklar uyuyordu.*
	'quand j'ai lu'		'Quand j'ai lu le livre, les enfants dormaient.'

TAB. 7.1 – Les trois fonctions du suffixe subordonnant *-DIK*

vu qu'il s'agît de formes verbales nominalisées. En effet, lorsqu'ils sont rattachés à des formes verbales non finies, les suffixes possessifs expriment l'accord avec le sujet : flexion nominale et flexion verbale se rapprochent ici jusqu'à se chevaucher, ce qui montre le statut tout particulier de ces formes.

7.2 Étiquetage des subordonnées non finies

La grammaire développée consiste en un total de vingt graphes repartis dans trois répertoires. Chaque catégorie comprend un graphe principal (en italique) et des sous-graphes, sauf la catégorie des relatives contenant seulement un graphe.

- PA : *AdvClause*, addition, agreement, concession, condition, information, manner, means, preference, proportion, purpose, quantity, reason, substitution, time ;
- PS : *NounClause*, NounClause_dik_ecek_nonSubj, NounClause_dik_ecek_Subj, NounClause_mek_me ;
- PR : *RelClause*.

Les graphes ne repèrent que les formes verbales régissant les SNF, la localisation des frontières des propositions étant un problème bien plus complexe qui requiert des informations syntaxiques non disponibles à ce niveau. Graphes et sous-graphes peuvent être utilisés pour simplement produire des concordances, ou bien en mode *MERGE* pour étiqueter les formes verbales par de balises XML.

Parmi les trois catégories listées, les converbes sont les formes verbales non finies les plus faciles à isoler pour deux raisons : (i) beaucoup d'entre eux contiennent des SS non ambigus et (ii) ceux qui contiennent des SS ambigus sont généralement marqués par un suffixe de cas spécifique

indique une action future.

et/ou suivis par une postposition particulière. Par exemple :

(i)	*-(y)IncA*,	(ii)	*-DIğIndA* (cas locatif),
	-(y)ArAk,		*-DIğI için/zaman/...* (postp.),
	-(y)ken		*-DIğIndAn beri* (cas ablatif plus postp.)

C'est pourquoi il a été décidé de développer la grammaire des propositions adverbiales en premier lieu. Les graphes des propositions substantives ont été écrits seulement après, et incluent la négation du graphe reconnaissant les PA. Enfin la grammaire des propositions relatives repose sur la négation des deux autres.

7.2.1 Propositions adverbiales

La grammaire des propositions adverbiales suit la division en catégories sémantiques faite par [Gök 05, pp. 473-485]. Le graphe principal (Fig. 7.1) consiste donc en l'union de quatorze sous-graphes. Pour des raisons de place, nous n'en présenterons ici qu'une partie.

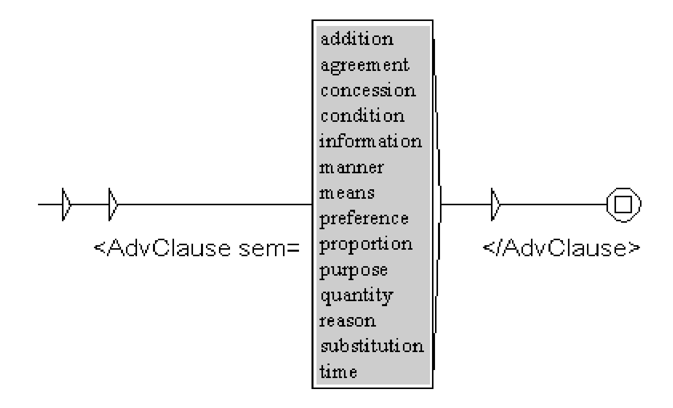


FIG. 7.1 – Grammaire des propositions adverbiales

La « concession » indique qu'il n'y a pas eu la relation logique attendue entre le fait exprimé par la subordonnée et celui exprimé par la principale[4] :

*[Kitabı **oku-ma-m-a rağmen**] onu hatirlamıyorum.*
'**Bien que j'aie lu** le livre, je ne m'en souviens pas.'

[4]Dans les exemples de ce chapitre, la subordonnée est notée entre crochets, la forme verbale non finie et l'éventuelle postposition sont notées en gras.

En turc les PA concessives (Fig. 7.2) sont régies par des SS ambigus, mais peuvent être reconnues grâces aux postpositions qui les suivent. Le premier des trois chemins du graphe comporte un SS sans marqueur de cas spécifique, le second demande le cas datif (*y*), tandis que le troisième requiert le suffixe d'accompagnement (***l*** *-lA*). Remarquons aussi que les deux premiers chemins impliquent la présence d'un suffixe possessif (*:1:2:3:4:5:6*), à la différence du dernier. Voici une partie des

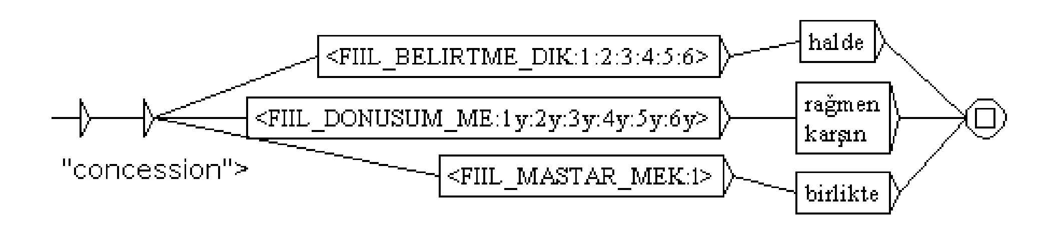


FIG. 7.2 – Grammaire des propositions adverbiales *concessives*

21 PA concessives repérées dans le corpus *RevOTAN* :

ütbeli generaller kalacakları **bilindiği halde** NATO'nun İngilizce Standardizasyon
nti-Semitik stereotiplemelere **dayanmasına rağmen**, İsrail'in Budapeşte, Prag, Var
rmiş ve örneğin sık sık davet **edilmelerine rağmen** Roma'daki NATO Savunma Kolejin
lışmaktalar. {S}İttifak üyesi **olmadıkları halde** ortak bir amaç doğrultusunda kan
de yapılan tartışmalar önemli **olmakla birlikte**, bu konudaki işbirliği "kağıt üze
yan iki nesil, kendi hataları **olmamasına rağmen** temel özgürlüklerinden mahrum ed

Les PA causales sont, elles aussi, régies par des SS ambigus. Elles prévoient toutes des postpositions spécifiques, exception faite pour la terminaison *-DIğIndAn/-(y)AcAğIndAn* après laquelle les postpositions *dolayı, ötürü* sont optionnelles.

*[Kitabı **oku-duğ-um-dan (dolayı)]** sorulara cevap verebildim.*
'J'ai su répondre aux questions **parce que j'ai lu** le livre.'

Le graphe les décrivant (Fig. 7.2) contient un « contexte », soit un dispositif intégré aux graphes syntaxiques d'Unitex qui permet d'en élever le pouvoir d'expression, depuis le niveau de grammaire algébrique (ou hors-contexte) à celui de grammaire contextuelle [Paum 06, pp. 94-95]. Dans notre cas le contexte est utilisé pour distinguer deux fonctions d'un même ensemble de formes verbales : en effet les terminaisons *-DIğIndAn/-(y)AcAğIndAn* (correspondant aux suffixes *-DIK* et *-(y)AcAK* suivis d'un possessif et du cas ablatif, cf. premier chemin du graphe) peuvent régir des PA causales mais aussi des propositions substantives.

*[Kitabı **oku-duğ-um-dan**] eminim.*
'Je suis sûr **que j'ai lu** le livre.'

Les PS construites sur une forme de type *-DIğIndAn/-(y)AcAğIndAn* ont valeur de complément indirect de verbes comme *kork-* ‘avoir peur (de)’, *emin ol-* ‘être sûr (de)’ et *şüphelen-* ‘douter (de)’. Pour éviter d’étiqueter ces occurrences comme PA, nous avons utilisé un contexte négatif, défini en délimitant une zone du graphe avec des boîtes contenant ‘$![’ et ‘$]’. Voici une partie des

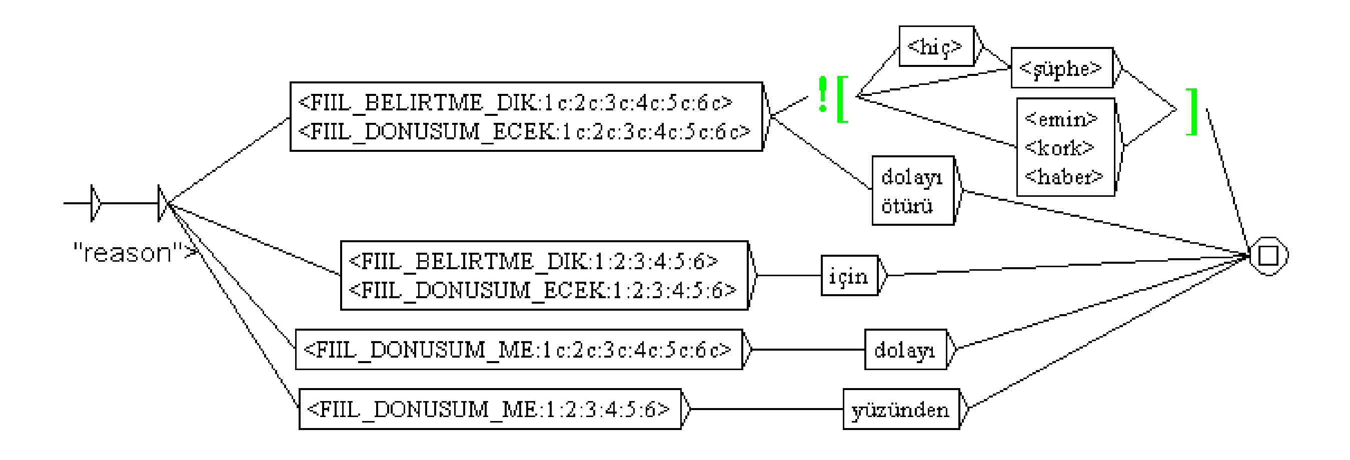


FIG. 7.3 – Grammaire des propositions adverbiales *causales*

69 PA causales repérées dans le corpus *RevOTAN* :

cazip gelecek mali olanaklar **bulunmadığından** yeterli niteliklere sahip yabancı
sistematik olarak gerçekleri **çarpıtmasından dolayı** Araplar komplo teorilerine b
İsrail’in yalnız kalacağından **korktuğu için**, Kudüs, Soğuk Savaş sırasında Batı y
yanlış davranışta bulunmaktan **korktukları için** seçeneklerini gereksiz yere kısıt
un rakipsiz olduğu bir konuda **odaklanacağı için** büyük ilgi ve destek gördü. {S}A
r.{S} Bugün kendimizi 1999’da **olduğumuzdan** çok daha az güvende olduğumuz konusun

Le plus complexe sous-graphe des PA, celui des propositions temporelles, est reporté à l’Annexe G. S’y référer aussi pour des exemples des converbes formés avec des SS non ambigus.

7.2.2 Propositions substantives

La grammaire des propositions substantives (Fig. 7.4) est divisée en trois parties selon le SS et le cas de la forme verbale non finie :

– *NounClause_mek_me* : décrit les PS construites sur des infinitifs (*-mAK*) ou sur des noms d’action généraux[5] (*-mA*),

[5] Le terme est emprunté à [Bazi 87, pp. 114-117].



FIG. 7.4 – Grammaire des propositions substantives

– *NounClause_dik_ecek_nonSubj* : décrits les PS régies par des formes verbales contenant *-DIK* ou *-(y)AcAK* et un suffixe de cas,
– *NounClause_dik_ecek_Subj* : décrits les PS régies par des formes verbales contenant *-DIK* ou *-(y)AcAK* mais pas de suffixe de cas.

Les trois sous-graphes commencent par un contexte contenant la négation de la grammaire des PA. Cela empêche que des cas spécifiques décrits par les graphes des PA, soient improprement reconnus comme PS aussi. Le suffixe *-(y)Iş* a été omis de la grammaire des PS, puisqu'il entraîne énormément de bruit. Ceci s'explique pour deux raisons : (i) le même suffixe peut jouer un rôle dérivationnel et est très productif, en tant que tel, dans la formation de noms abstraits ou semi-abstraits (ex. *giriş* 'entrée', *kuruluş* 'fondation') [Gök 05, p. 427] et (ii) il n'est pas facilement distinguable du suffixe de voix réciproque *-Iş*. Voilà un cas où information morphologique et contexte se révèlent insuffisants à la reconnaissance de la subordination.

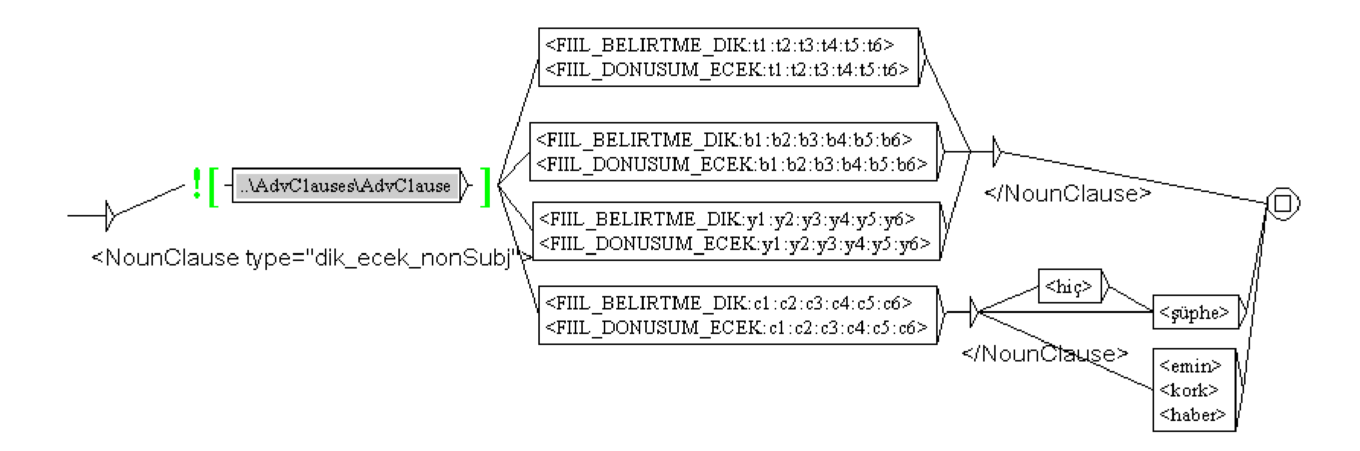


FIG. 7.5 – Grammaire des propositions substantives *-DIK/-(y)AcAk non sujet*

Le graphe *NounClause_dik_ecek_nonSubj* (Fig. 7.5) reconnaît des formes verbales au cas génitif (*t*), accusatif (*b*) et datif (*y*) sans restrictions, ainsi que des formes ablatives (*c*) suivies d'un ensemble restreint de verbes et de substantifs (présentés à la sous-section précédente). Dans ces cas la PS a valeur de complément direct ou indirect d'un verbe ou d'un substantif de la proposition principale.

*[Kitabı **oku-duğ-um-u**] biliyor.*
'Il sait **que j'ai lu** le livre.'

*[Kitabı **oku-duğ-um-a**] inaniyor.*
'Il croit **(au fait) que j'ai lu** le livre.'

*[Kitabı **oku-duğ-um-un**] farkında oldu.*
'Il s'est aperçu **(du fait) que j'ai lu** le livre.'

Le graphe *NounClause_dik_ecek_nonSubj* appliqué au corpus *RevOTAN* reconnaît 377 formes verbales, dont voici une partie :

antik Konseyi "26 üye" ile de **çalışabileceğini** kanıtladı.{S} Çek Cumhuriyeti'nin
şmek için sınırlarının dışına **çıkacağının** habercisidir.{S} Afganistan'daki ISAF
aranti edecek bir planla geri **döneceğini** bildirmesiyle sona erdi." {S}Tabi bir p
lerde devrim" hamlesine dahil **edebileceğini** düşündüğü bir çok programın maliyeti
etenekler, yetenekler" olması **gerektiğini** tekrarlamaya başladı. {S}Ancak Kosova'
if bir yaklaşıma sahip olması **gerektiğini** iddia etmektedir.{S} Belge, transatlan
ilgili reformların boşa gidip **gitmeyeceğini** sorguluyordu.{S} Bu anlamda sadece G
arda 240 kereden fazla kuvvet **kullandığını** ve söz konusu durumların dörtte üçünd
alar bile ilerlemenin çok zor **olduğuna** işaret ediyor.{S} Özellikle NATO Güvenlik
konusunun Avrupa'nın dışında **olduğunu** da yansıtmaktadır :{S} Kore'nin güvenliği

Mais les PS peuvent aussi jouer le rôle de sujet d'un verbe passif ou d'un prédicat nominal :

*[Kitabı **oku-duğ-um**] bilinir.*
litt. '**(Le fait) que j'ai lu** le livre est su.' -> 'On sait **que j'ai lu** le livre.'

*[Kitabı **oku-duğ-um**] doğru.*
litt. '**(Le fait) que j'ai lu** le livre est vrai.' -> 'Il est vrai **que j'ai lu** le livre.'

Puisque la syntaxe du dictionnaire *inflex* ne permet pas d'exprimer le cas nominatif, correspondant à $\emptyset$-suffixe, dans le graphe des PS sujet (Fig. 7.5) nous utilisons la négation de *NounClause_dik_ecek_nonSubj* afin de repérer les formes au nominatif.

Comme l'on observe dans le graphe, le contexte droite des formes est restreint pour qu'elles puissent être distinguées des mêmes formes ayant fonction de proposition relative. Il reste pourtant du bruit parmi les 224 formes repérées par le graphe *NounClause_dik_ecek_Subj* dans le corpus *RevOTAN* :

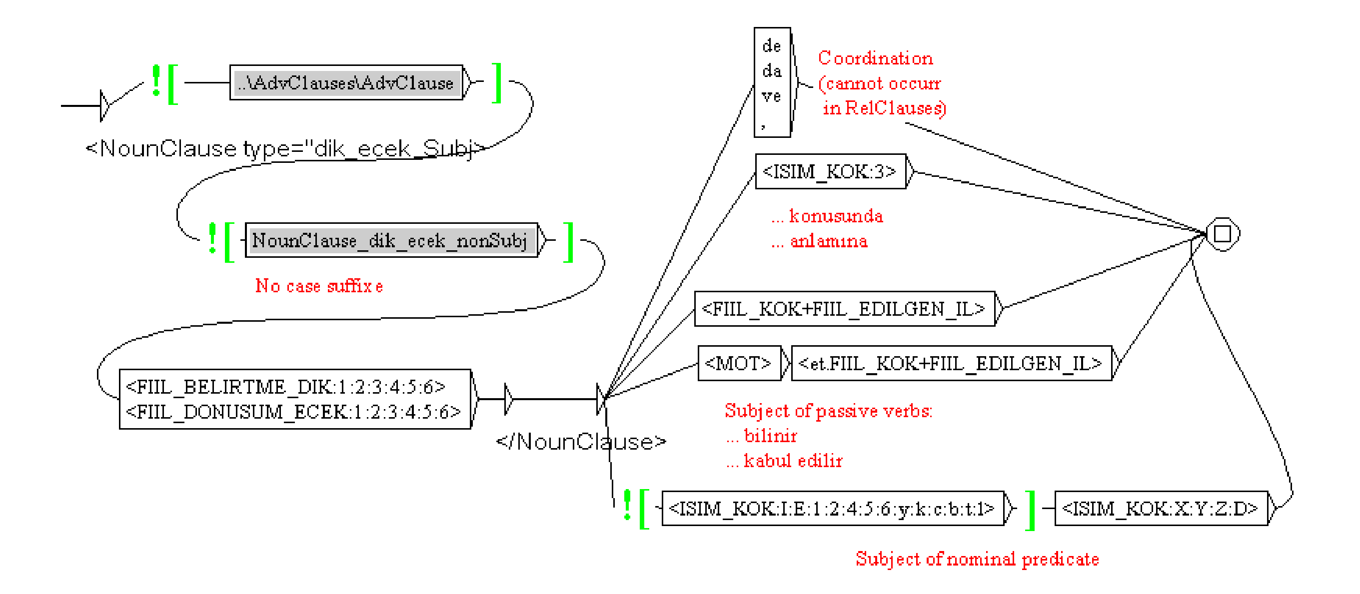


FIG. 7.6 – Grammaire des propositions substantives *-DIK/-(y)AcAk sujet*

önceliklerin ve ihtiyaçların **bulunduğu alanlardır**.{S} Bir çok uzmanın korktuğu
kimin planlayacağı ve kontrol **edeceği konusunda** Avrupa Birliği ve NATO arasında
işimlere bir reaksiyon olarak **geliştiği görülebilir** :{S} Londra Zirvesi, Soğuk Sa
rupalıların kendilerini rahat **hissettikleri ve** gerçekçi olarak katkıda bulunabil
i ile bir arada operasyonlara **katılabilecekleri garanti edilecektir**. {S}Askeri d
yeni operasyonel kavramların **kullanılabileceği dönüştürülmüş** bir askeri varlık
n eylemlere kolaylıkla adapte **olabildikleri görülmüştür**.{S} Mevcut savaş uçaklar
reksinimlerinin de değişmekte **olduğu açıktır**.{S} Asya ve Orta Doğu'da hala öneml
ok uzaklardaki olaylara bağlı **olduğu anlamına** geliyor.{S} İttifak, Prag'da, bu d
abalarında yıllardır yapmakta **olduğu reformları** hızlandırma fırsatı vermiştir.{S
inasyon için bir forum olarak **oynadığı rolü** güçlendirmeyi" taahhüt ettiler. (NA

Beaucoup de ces formes sont bien des PS sujet, mais on y trouve aussi des PR, la plupart des erreurs étant dues à l'impossibilité de distinguer le suffixe de l'accusatif de celui du possessif à la 3e personne du singulier. Ainsi la boîte [ISIM_KOK :3] peut reconnaître erronément un nom à l'accusatif qui est en fait le constituant modifié par la PR. C'est le cas des deux dernières formes listées :

abalarında yıllardır yapmakta **olduğu reformları** hızlandırma fırsatı vermiştir.{S
inasyon için bir forum olarak **oynadığı rolü** güçlendirmeyi" taahhüt ettiler. (NA

Le graphe des PS construites sur les suffixes *-mAK* et *-mA* est donné à l'Annexe G.

7.2.3 Propositions relatives

Les propositions relatives, elles aussi, se divisent selon le SS qui les forme :

- le SS non ambigu -(y)An indique généralement que le constituant relativisé est le sujet du verbe subordonné,

  *[Kitabı **oku-yan**] adam gülümsüyor.*
  'L'homme **qui lit** le livre sourit.'

- les SS ambigus *-DIK* et -(y)AcAk peuvent masquer diverses relations syntaxiques entre le constituant relativisé et le sujet du verbe subordonné (objet direct et indirect, modifieur adverbial, possession). Dès lors, ils sont à l'origine des SNF les plus difficiles à analyser :

  *[**Oku-duğ-um**] kitabı kaybettim.*
  'J'ai égaré le livre **que j'ai lu**.'

  *[Kitabı **oku-duğ-um**] gözlüğü kaybettim.*
  'J'ai égaré les lunettes **avec lesquelles j'ai lu** le livre.'

  *[Kitabını **oku-duğ-um**] yazarla tanıştım.*
  'J'ai rencontré l'écrivain **dont j'ai lu** le livre.'

Les deux types de PR peuvent être visualisés dans le graphe à la Fig. 7.7. Encore une fois nous nous servons d'un contexte négatif pour imposer que la forme verbale recherchée soit au nominatif. En effet les formes verbales construites avec *-DIK/-(y)AcAk* contenant un suffixe de cas sont incluses dans le sous-graphe des PS non sujet. En niant la grammaire des PS, nous essayons ainsi d'isoler les formes au nominatif. Selon notre grammaire, le corpus *RevOTAN* contient 2033

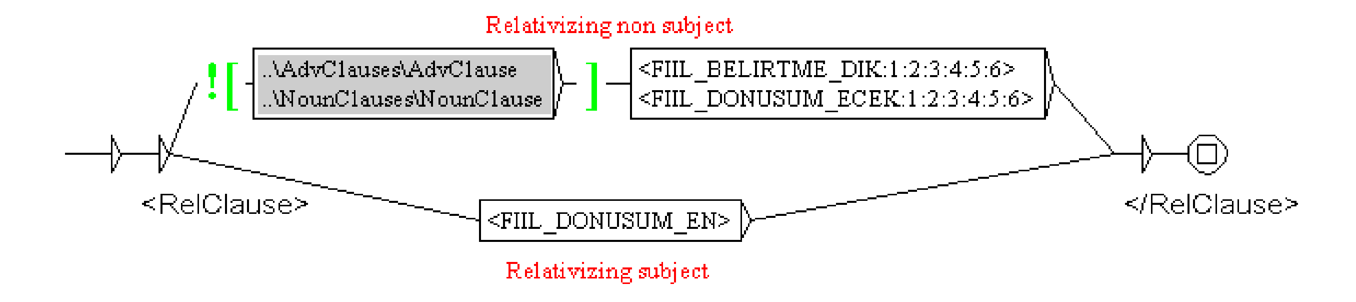


FIG. 7.7 – Grammaire des propositions relatives

PR. Le bruit y est sûrement important et mériterait d'être examiné de plus près. Ci-dessous un extrait de la concordance générée :

a başladı. {S}Ancak Kosova'da **alınan** dersler daha henüz özümsenirken NATO'nun st
utum meselesi. {S}ABD Savunma **Bakanı** Donald H. Rumsfeld {S}Ronald Reagan'ın başk
11 Eylül dersleri" karşısında **başlatılan** ve yetenekler, misyonlar, ve yapılar üz
Bakanı Donald H. Rumsfeld'in **geçenlerde** ifade ettiği gibi, dönüşüm bir sonuçtan
perasyonlarımızın ve NATO'nun **geleceği** üzerinde asılı kalacak" demişti. {S}NATO
üz biçimde işleyebilmesi için **gereken** prosedürleri onayladı. {S}Geçen iki buçuk
değişim, daha çok adaptasyon **gerektirdiği** bir süreç. "Dönüşüm" sözcüğü son yıll
mini hem de yaşayabilirliğini **kaybedeceği** şeklindeki uyarılara cevap vermek zoru
rına bir saldırıda bulunmamış **olan** bir devlete (dağılmakta olan Yugoslavya) karş
a ve karar alma mekanı olarak **oynadığı** roldeki değişiklikler üzerinde odaklanan
tik tartışma :{S} Şubat ayında **yaptıkları** toplantıda İttifak liderleri "NATO'nun
üm 21. yüzyılda dünyanın bizi **zorladığı** bir süreç, ve teknolojiler ve platformla

7.3 Discussion

Les résultats observés sont encourageants. Il faut toutefois tenir compte du fait que les graphes ont été développés en lien étroit avec les corpus de travail (*RevOTAN* en particulier). En d'autres mots, l'écriture de la grammaire a été influencée par les occurrences observées dans les corpus dont nous disposions. Il faudrait donc en vérifier la précision et le rappel par l'application à d'autres échantillons de textes.

Nous avons tout de même réussi à produire un outil d'étiquetage des verbes régissant des propositions subordonnées non finies, en exploitant l'information morphologique et le contexte des formes verbales recherchées. Nous avons ainsi démontré que, dans des cas délimités, il est possible d'enrichir le texte par des annotations linguistiques de haut niveau, sans devoir passer par l'analyse syntaxique, coûteuse et souvent indisponible.

Il s'agît, à notre avis, d'un genre d'outil très précieux car peu demandant en ressources linguistiques et logicielles. Nous croyons que cette démarche pourrait également profiter à la recherche dans le TAL basé sur des statistiques. D'ailleurs, l'outil développé peut avoir des applications immédiates, comme par exemple la visualisation en contexte de structures linguistiques complexes posant des problèmes aux apprenants du turc langue étrangère.

Conclusion

L'objectif fondamental de l'ouvrage, soit l'intégration du turc aux langues d'Unitex, a été atteint. Les caractéristiques morphologiques du turc ont donné une ligne directive à tout notre travail, de sa conception jusqu'à son évaluation et à son application.

Après avoir choisi l'analyseur linguistique qu'aurait constitué la ressource de départ, nous nous y sommes familiarisés et avons conduit, par la force des choses, un exercice de terminologie turco-anglais-français sans lequel il aurait été difficile de manipuler un outil entièrement programmé et documenté en turc.

Lors de l'analyse des avantages et des limites de chaque solution envisagée, aucune méthode n'est émergée comme ayant un avantage clair du point de vue théorique. Nous sommes ainsi parvenus à la conclusion qu'une seule forme de représentation linguistique ne pouvait être suffisante et avons décidé d'en implémenter deux. D'un côté, le dictionnaire *inflex* est la ressource essentielle de tout traitement de corpus que l'on souhaite effectuer dans Unitex. De l'autre, la génération de l'automate du texte explicitant les ambiguïtées morphologiques ouvre la voie au développement de grammaires locales de levée des ambiguïtées, qui serait à notre avis une application très prometteuse d'Unitex à la langue turque.

Il serait impropre d'affirmer que nous avons créé une nouvel outil de TAL. Néanmoins nous estimons que, grâce à la transformation et à l'intégration raisonnées de ressources existantes, des nouveaux instruments de travail ont été mis à la disposition des chercheurs, linguistes ou ingénieurs, ainsi que des apprenants de la langue turque. Les cas pratiques d'application d'Unitex au turc présentés en fin d'ouvrage (recherche de motif, grammaire de classification des SNF) semblent confirmer notre propos.

Des perspectives s'ouvrent donc dans deux directions : tout premièrement l'optimisation du modèle de représentation conçu, et en suite l'avancement des recherches sur l'étiquetage et l'analyse de la subordination. Nous comptons poursuivre le travail fait jusqu'ici en nous nous souhaitant d'avoir contribué quelque peu au domaine des études de linguistique de corpus de la langue turque.

Bibliographie

[Akı] A. A. AKIN AND M. D. AKIN. *Zemberek, an open source NLP framework for Turkic Languages*.

[Aleg 08] I. ALEGRIA, K. CEBERIO, N. EZEIZA, A. SOROA, AND G. HERNANDEZ. “Spelling Correction : from Two-Level Morphology to Open Source”. In : *Proceedings of the Sixth International Language Resources and Evaluation (LREC’08)*, European Language Resources Association (ELRA), Marrakech, Morocco, may 2008.

[Antw 93] E. ANTWORTH. “Glossing text with the PC-KIMMO morphological parser”. *Computers and the Humanities*, Vol. 26, pp. 389–98, 1993.

[Antw 97] E. ANTWORTH AND S. MCCONNEL. *PC-Kimmo Reference Manual : a two-level processor for morphological analysis*. version 2.1.0 Ed., October 1997.

[Aydi 07] ÖZGÜR AYDIN. “The comprehension of Turkish relative clauses in second language acquisition and agrammatism”. *Applied Psycholinguistics*, Vol. 28, pp. 295–315, 2007.

[Bazi 87] L. BAZIN. *Introduction à l’étude pratique de la langue turque*. Librairie d’Amérique et d’Orient, 1987.

[Cagr 05] I. CAGRI. “Acquisition of Turkish Relative Clauses”. 2005. Unpublished manuscript.

[Calb 03] M. CALBERG. “Traitement de la morphologie du finnois par transducteurs à nombre fini d’états”. In : *RÉCITAL 2003, Batz-sur-Mer*, 2003.

[Eryi 04] G. ERYIĞIT AND E. ADALI. “An Affix Stripping Morphological Analyzer for Turkish”. In : *Proceedings of the IASTED International Conference Artificial Intelligence and Application*, 2004.

[Gros 86] M. GROSS. “Lexicon Grammar. The Representation of Compound Words”. In : *COLING*, pp. 1–6, 1986.

[Gök 05] A. GÖKSEL AND C. KERSLAKE. *Turkish : a comprehensive grammar*. Routledge, 2005.

[Kosk 83] K. KOSKENNIEMI. *Two-level Morphology : A General Computational Model for Word-Form Recognition and Production*. University of Helsinki, Department of General Linguistics, 1983.

[Lapo 98] E. LAPORTE AND A. MONCEAUX. "Elimination of lexical ambiguities by grammars : The ELAG system". *Lingvisticæ Investigationes*, Vol. 22, p. 341–367, 1998.

[Ofla 93] K. OFLAZER. "Two-level description of Turkish Morphology". In : *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, 1993.

[Paum 06] S. PAUMIER. *UNITEX 1.2 Manuel d'utilisation*. Université de Marne-la-Vallée, 2006.

[Silb 94] M. D. SILBERZTEIN. "INTEX : a corpus processing system". In : *Proceedings of the 15th conference on Computational linguistics*, pp. 579–583, Association for Computational Linguistics, Morristown, NJ, USA, 1994.

[Özt 96] S. M. ÖZTANER. *A Word Grammar of Turkish with Morphophonemic Rules*. Master's thesis, The Graduate School of Natural and Applied Sciences of the Middle East Technical University, 1996.

Annexe A

Texte turc analysé par Zemberek

Ce qui suit est un exemple d'analyse produite par la classe de test `DemoMain`. Chaque mot du texte analysé est suivi de ses possibles analyses : entre crochets la racine (*kok*) du mot suivie de sa catégorie, puis les noms des suffixes (*ekler*) repérés. Dans la première colonne la segmentation en racine et suffixes, dans la seconde la catégorie de la racine et les traits grammaticaux associés aux suffixes.

```
İnsan
[ Kok: insan, ISIM ]
hakları
[ Kok: hak, ISIM ] Ekler: ISIM_COGUL_LER + ISIM_BELIRTME_I
[ Kok: hak, ISIM ] Ekler: ISIM_COGUL_LER + ISIM_SAHIPLIK_O_I
[ Kok: hak, ISIM ] Ekler: ISIM_SAHIPLIK_ONLAR_LERI
evrensel
[ Kok: evren, ISIM ] Ekler: ISIM_ILISKILI_SEL
beyannamesi
[ Kok: beyanname, ISIM ] Ekler: ISIM_TAMLAMA_I
[ Kok: beyanname, ISIM ] Ekler: ISIM_SAHIPLIK_O_I
Önsöz
:cozulemedi
İnsanlık
[ Kok: insan, ISIM ] Ekler: ISIM_BULUNMA_LIK
[ Kok: insan, ISIM ] Ekler: ISIM_DURUM_LIK
ailesinin
[ Kok: aile, ISIM ] Ekler: ISIM_TAMLAMA_I + ISIM_TAMLAMA_IN
[ Kok: aile, ISIM ] Ekler: ISIM_SAHIPLIK_O_I + ISIM_TAMLAMA_IN
bütün
[ Kok: bütün, ISIM ]
üyelerinde
[ Kok: üye, ISIM ] Ekler: ISIM_COGUL_LER + ISIM_TAMLAMA_IN + ISIM_KALMA_DE
[ Kok: üye, ISIM ] Ekler: ISIM_COGUL_LER + ISIM_SAHIPLIK_O_I + ISIM_KALMA_DE
[ Kok: üye, ISIM ] Ekler: ISIM_COGUL_LER + ISIM_SAHIPLIK_SEN_IN + ISIM_KALMA_DE
[ Kok: üye, ISIM ] Ekler: ISIM_SAHIPLIK_ONLAR_LERI + ISIM_KALMA_DE
```

Annexe B

Glossaire des suffixes Zemberek

Les noms de suffixes Zemberek sont toujours notés en majuscule et sans diacritiques. Ils sont formés de trois éléments unis par le caractère '_'. Font exception les marqueurs de personne, qui en contiennent quatre :

<categ>_<nom>_<forme>
<categ>_<nom>_<personne>_<forme>

Le premier élément *<categ>* désigne la catégorie de la racine à la quelle le suffixe peut être rattaché : *ISIM* nom, *FIIL* verbe, *OZEL* nom propre, *SIFAT* adjectif, *BAGLAC* conjonction, *EDAT* clitique, *SAYI* nombre.

Le second *<nom>* est la véritable appellation du suffixe. Le même nom peut être partagé par plusieurs suffixes, auquel cas ils seront distingués par leur forme de surface canonique *<forme>*. Dans le cas des marqueurs de personne, celle-ci est renseignée juste avant la forme, en troisième position *<personne>* : *BEN* je, *SEN* tu, *O* il/elle, *BIZ* nous, *SIZ* vous, *ONLAR* ils/elles.

Nous donnons ici la liste des noms de suffixes apparaissant dans ce mémoire.

- Suffixes de cas :

YONELME	datif
KALMA	locatif
CIKMA	ablatif
BELIRTME	accusatif
TAMLAMA	génitif ou marqueur de groupe nominal

- Autres suffixes de flexion nominale :

COGUL	pluriel
SAHIPLIK	possessif
BULUNMA_KI	suffixe pronominal textit-ki

- Voix verbales :

EDILGEN	passive
ETTIRGEN	causative
BERABERLIK	réciproque

- Modes/modalités/temps verbaux :

GENISZAMAN	aoriste
GECMISZAMAN	passé
ZORUNLULUK_MELI	obligatif
YETENEK_EBIL	possibilité, capacité

- Noms verbaux d'action (suffixes subordonnants) :

BELIRTME_DIK	nom verbal complexe de réalité
DONUSUM_ME	nom d'action général
MASTAR_MEK	infinitif

- Autres suffixes de flexion verbale :

KISI	accord verbal de personne
OLUMSUZLUK	négation

- Dérivation :

BULUNMA_LI	nom -> nom, adjectif (possession, origine)
DONUSUM_LE	nom -> verbe

Annexe C

Table de conversion des suffixes flexionnels

Ceci est le contenu du fichier *inflectionalCodes.txt* placé dans le répertoire *kaynaklar/toUnitex* ajouté à la bibliothèque Zemberek.

```
# Nominal inflectional suffixes
# (lower case codes)
# Plural -ler
ISIM_COGUL_LER p

# Possessive
ISIM_SAHIPLIK_BEN_IM 1
ISIM_SAHIPLIK_SEN_IN 2
ISIM_SAHIPLIK_O_I 3
ISIM_SAHIPLIK_BIZ_IMIZ 4
ISIM_SAHIPLIK_SIZ_INIZ 5
ISIM_SAHIPLIK_ONLAR_LERI 6

# Case
ISIM_YONELME_E y
ISIM_KALMA_DE k
ISIM_CIKMA_DEN c
ISIM_BELIRTME_I b
ISIM_TAMLAMA_IN t

# -(y)le
ISIM_BIRLIKTELIK_LE l
####################################
# Verbal inflectional suffixes
# (upper case codes)

# Negation
FIIL_OLUMSUZLUK_ME N
```

```
# Tenses/Aspect/Modality
# Futur, Perfective, Evidential, Aorist, Imperfective
FIIL_GELECEKZAMAN_ECEK F
FIIL_GECMISZAMAN_DI P
FIIL_GECMISZAMAN_MIS E
FIIL_GENISZAMAN_IR A
FIIL_SIMDIKIZAMAN_IYOR I

# Conditional (Hypothetical), Obligative, Optative (Wish)
FIIL_SART_SE H
FIIL_ZORUNLULUK_MELI O
FIIL_ISTEK_E W

# Copular markers
IMEK_HIKAYE_DI X
IMEK_RIVAYET_MIS Y
IMEK_SART_SE Z

# Person markers
# Verbal
FIIL_KISI_BEN 1
FIIL_KISI_SEN 2
FIIL_KISI_O 3
FIIL_KISI_BIZ 4
FIIL_KISI_SIZ 5
FIIL_KISI_ONLAR 6

# Nominal predicates
ISIM_KISI_BEN_IM 1
ISIM_KISI_SEN_SIN 2
ISIM_KISI_O_BOS 3
ISIM_KISI_BIZ_IZ 4
ISIM_KISI_SIZ_SINIZ 5
ISIM_KISI_ONLAR_LER 6

# Imperative
# iMperative
FIIL_EMIR_O_SIN 3M
FIIL_EMIR_SIZ_IN 5M
FIIL_EMIR_SIZRESMI_INIZ 5M
FIIL_EMIR_ONLAR_SINLER 6M

FIIL_ISTEK_SENE 2M
FIIL_ISTEK_SENIZE 5M

# Generalizing modality marker
ISIM_TANIMLAMA_DIR D
```

Annexe D

Extrait du dictionnaire *inflex* du corpus *Fables*

Nous reportons ici une partie du dictionnaire *inflex* du corpus *Fables* généré par le package `toUnitex` en utilisant l'analyse morphologique de Zemberek. Le dictionnaire n'est pas trié, cette opération étant assurée par Unitex au moment du chargement.

```
deyince,deyi.ISIM_KOK+ISIM_SAHIPLIK_SEN_IN+ISIM_TARAFINDAN_CE
deyince,de.FIIL_KOK+FIIL_ZAMAN_INCE
değil,değil.ISIM_KOK
değil,değ.FIIL_KOK+FIIL_EDILGEN_IL
birer,birer.ISIM_KOK
birer,bir.SAYI_KOK+SAYI_ULESTIRME_ER
bir,bir.SAYI_KOK
sözleri,söz.ISIM_KOK:pb:p3:6
kralı,kral.ISIM_KOK:3:b
kaşla,kaş.ISIM_KOK+ISIM_DONUSUM_LE
kaşla,kaş.ISIM_KOK:l
konuştuklarını,konuş.FIIL_KOK+FIIL_BELIRTME_DIK:p3b:p2b:6b
konuştuklarını,kon.FIIL_KOK+FIIL_BERABERLIK_IS+FIIL_BELIRTME_DIK:p3b:p2b:6b
buğdaylar,buğday.ISIM_KOK:p:6
buğdaylar,buğday.ISIM_KOK+ISIM_DONUSUM_LE:A
edip,edip.ISIM_KOK
edip,et.FIIL_KOK+FIIL_IMSI_IP
çağırıyor,çağır.FIIL_KOK:I
bela,bela.ISIM_KOK
inmiş,in.FIIL_KOK:E
inmiş,in.FIIL_KOK+FIIL_DONUSUM_MIS
inmiş,in.ISIM_KOK:Y
karşılıksız,karşılık.ISIM_KOK+ISIM_YOKLUK_SIZ
karşılıksız,karşı.ISIM_KOK+ISIM_BULUNMA_LIK+ISIM_YOKLUK_SIZ
açın,açın.FIIL_KOK
açın,açı.ISIM_KOK:2
açın,aç.FIIL_KOK:5M
```

```
açın,aç.ISIM_KOK:t:2
gözü,göz.ISIM_KOK:3:b
acımalı,acı.FIIL_KOK+FIIL_DONUSUM_ME+ISIM_BULUNMA_LI
acımalı,acı.FIIL_KOK:O
koşmuş,koş.FIIL_KOK:E
koşmuş,koş.FIIL_KOK+FIIL_DONUSUM_MIS
güvenmeli,güven.FIIL_KOK+FIIL_DONUSUM_ME+ISIM_BULUNMA_LI
güvenmeli,güven.FIIL_KOK:O
dönen,dönen.FIIL_KOK
dönen,dön.FIIL_KOK+FIIL_DONUSUM_EN
yapmış,yap.FIIL_KOK:E
yapmış,yap.FIIL_KOK+FIIL_DONUSUM_MIS
söyleyen,söyle.FIIL_KOK+FIIL_DONUSUM_EN
herkes,herkes.ZAMIR_KOK
tamam,tamam.ISIM_KOK
gün,gün.ZAMAN_KOK
süre,süre.ISIM_KOK
süre,sür.FIIL_KOK:W
süre,sür.FIIL_KOK+FIIL_YETERSIZLIK_E
canlanmış,canlan.FIIL_KOK:E
canlanmış,canlan.FIIL_KOK+FIIL_DONUSUM_MIS
canlanmış,can.ISIM_KOK+ISIM_DONUSUM_LE+FIIL_EDILGENSESLI_N:E
canlanmış,can.ISIM_KOK+ISIM_DONUSUM_LE+FIIL_EDILGENSESLI_N+FIIL_DONUSUM_MIS
herkes,herkes.ZAMIR_KOK
sonunda,son.SAYI_KOK:2k:3k
su,su.ISIM_KOK
felaketler,felaket.ISIM_KOK:p:6
felaketler,felaket.ISIM_KOK+ISIM_DONUSUM_LE:A
sahibimin,sahip.ISIM_KOK:1t
canına,can.ISIM_KOK:3y:2y
sözlerine,söz.ISIM_KOK:p3y:p2y:6y
kapacak,kap.FIIL_KOK:F
kapacak,kap.FIIL_KOK+FIIL_DONUSUM_ECEK
savaşırlar,savaş.FIIL_KOK:A6
dışarı,dışarı.ISIM_KOK
kuşku,kuşku.ISIM_KOK
tarla,tarla.ISIM_KOK
tarla,tar.ISIM_KOK+ISIM_DONUSUM_LE
tarla,tar.ISIM_KOK:l
silahlanıp,silahlan.FIIL_KOK+FIIL_IMSI_IP
silahlanıp,silahla.FIIL_KOK+FIIL_EDILGENSESLI_N+FIIL_IMSI_IP
silahlanıp,silah.ISIM_KOK+ISIM_DONUSUM_LE+FIIL_EDILGENSESLI_N+FIIL_IMSI_IP
komşularına,komşu.ISIM_KOK:p3y:p2y:6y
```

Annexe E

Automate d'une phrase turque

Généré avec Zemberek + package toUnitex Visualise les segmentations possibles en morphèmes

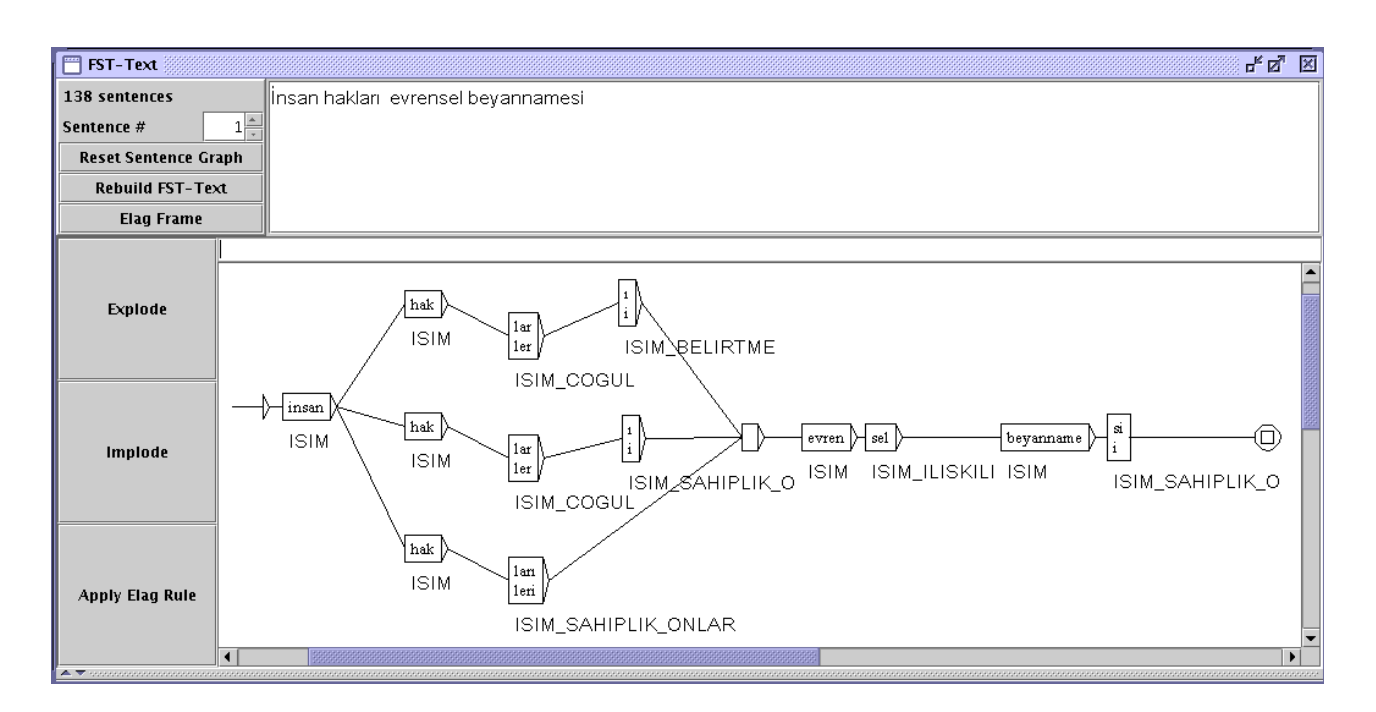


FIG. E.1 – Automate d'une phrase turque

Annexe F

Le package *toUnitex* : extraits de code JAVA

Génération de codes flexionnels.

La méthode ***DelaEntry*** de la classe `ZemberekToUnitex` génère les entrées du dictionnaire *inflex*. L'algorithme parcourt la séquence de suffixes d'une forme de droite à gauche et génère un ensemble de codes flexionnels au travers de la table de conversion. Les suffixes qui ont été trouvés dans la table de conversion sont retirés, en même temps, de la séquence d'informations sémantiques-grammaticales.

```
public String DelaEntry(String word, Kelime k){

    String entry = "";
    entry += toLowerCase(word,k) + "," + k.kok().icerik() + ".";
    List<Ek> suffixes = k.ekler();
    Stack<String> flexCodes = new Stack<String>();

    // Analyze suffixes backward since inflectional suffixes must
    // always come at the end
    for(int i=suffixes.size()-1; i>=0; i--){
       if(inflectionalCodes.containsKey(suffixes.get(i).ad())){
          flexCodes.add(inflectionalCodes.get(suffixes.get(i).ad()));
          suffixes.remove(i);
       }
       else break;
    }

    // always write first suffix, which is actually the category
    // of the root
    entry += suffixes.get(0).ad();
    // write following suffixes if any
    for(int j=1; j<suffixes.size(); j++){
       entry += "+" + suffixes.get(j).ad();
    }

    // write inflectional codes if any
    if(flexCodes.size()!=0) {
       entry += ":";
       while(!flexCodes.isEmpty()){
          String flexCode = flexCodes.pop();
```

```
          entry += flexCode;
       }
    }
    return entry;
}
```

Factorisation d'entrées similaires.

La classe **`WordDelaEntries`** est utilisée par la méthode de génération d'entrées *`DelaEntry`* pour gérer la factorisation d'entrées similaires. Une instance de cette classe représente l'ensemble d'entrées DELAF associées à une forme.

```
package toUnitex;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

/**
 * @author arianna
 * This class represents a set of DELA entries corresponding
 * to different analysis of a Turkish word.
 * It handles factorisation of similar entries.
 * Two entries are similar if they differ only in their
 * flexional field.
 *
 */
public class WordDelaEntries extends ArrayList<String> {

   public boolean add(String entry){
      // first check if an identical entry already exists
      if(this.contains(entry)) {return true};
      // then look for similar entries
      int indexSimEntry = this.indexOfSimilarEntry(entry);
      if(indexSimEntry == -1) {
         super.add(entry);
      }
      else {
         int comma = entry.indexOf(':');
         if(comma == -1) {return true; }
         String newEntry = this.get(indexSimEntry) + ":" + entry.substring(comma+1);
         this.set(indexSimEntry, newEntry);
      }
      return true;
   }

    /**
        * Return index of the first similar entry.
        *
        * @param entry entry to be compared
        * @return index of the first similar entry, or -1 if no similar entry is found
        */
   public int indexOfSimilarEntry(String entry) {
      for(int i=0; i<this.size(); i++) {
         if(stripInflexionalCodes(this.get(i)).equals(stripInflexionalCodes(entry)))
            {return i; }
      }
```

```
      return -1;
   }

   public String stripInflexionalCodes(String entry) {
      String beforeInflexCodes = null;
      int comma = entry.indexOf(':');
      if(comma==-1) {beforeInflexCodes = entry;}
      else {beforeInflexCodes = entry.substring(0, comma);}
      return beforeInflexCodes;
   }

}
```

Annexe G

Graphes d'étiquetage des subordonnées non finies

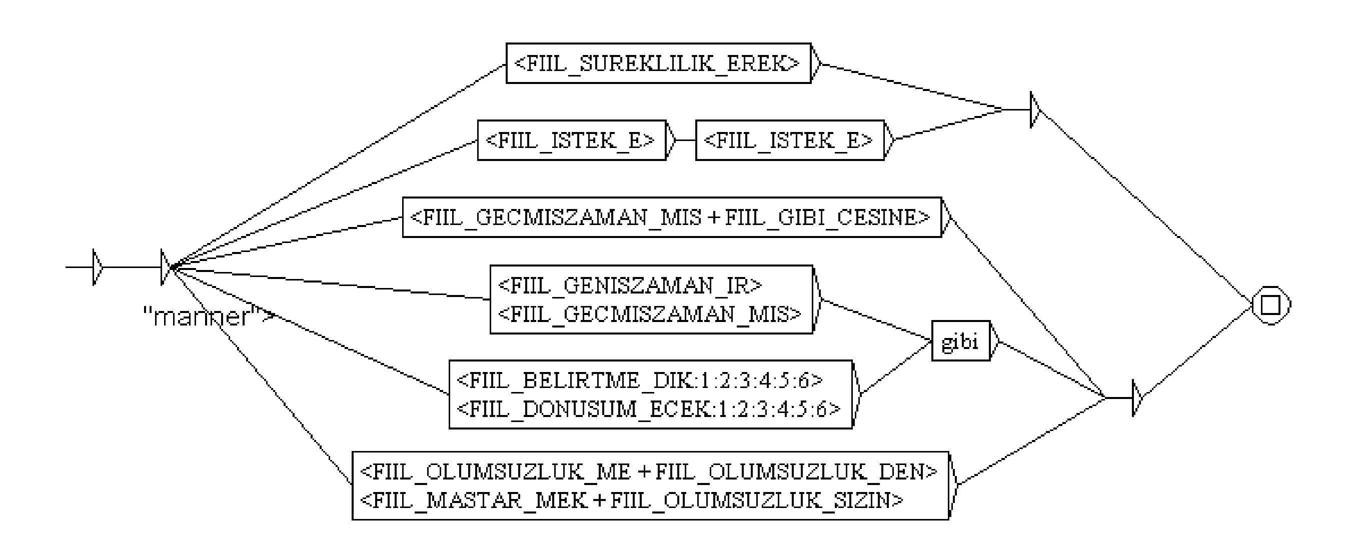


FIG. G.1 – Grammaire des propositions adverbiales *de manière*

<FIIL_ZAMAN_INCE>
<FIIL_DONUSUM_ME-ISIM_KOK:1:2:3:4:5:6>
üzerine
tr="when">
<FIIL_BELIRTME_DIK:1:2:3:4:5:6>
zaman
sırada
<FIIL_BELIRTME_DIK:1k:2k:3k:4k:5k:6k>
<FIIL_BELIRTME_DIK:1:2:3:4:5:6>
anda
<FIIL_GENISZAMAN_IR>
<FIIL_OLUMSUZLUK_ME+FIIL_GENISZAMAN_IR>
tr="asSoonAs">
<FIIL_GENISZAMAN_IR+IMEK_ZAMAN_KEN>
<ISIM_KOK+IMEK_ZAMAN_KEN>
"time"
tr="while">
<FIIL_BELIRTME_DIK:1c:2c:3c:4c:5c:6c>
beri
tr="since">
<FIIL_OLUMSUZLUK_ME+FIIL_OLUMSUZLUK_DEN>
önce
tr="before">
<FIIL_BELIRTME_DIK:c>
sonra
tr="after">
<FIIL_DONUSUM_ME:1c:2c:3c:4c:5c:6c>
itibaren
<FIIL_ZAMAN_INCE:y>
kadar
değin
dek
tr="until">
<FIIL_BELIRTME_DIK:1:2:3:4:5:6>
sürece
müddetçe
tr="throughoutTheTime">

FIG. G.2 – Grammaire des propositions adverbiales *temporelles*

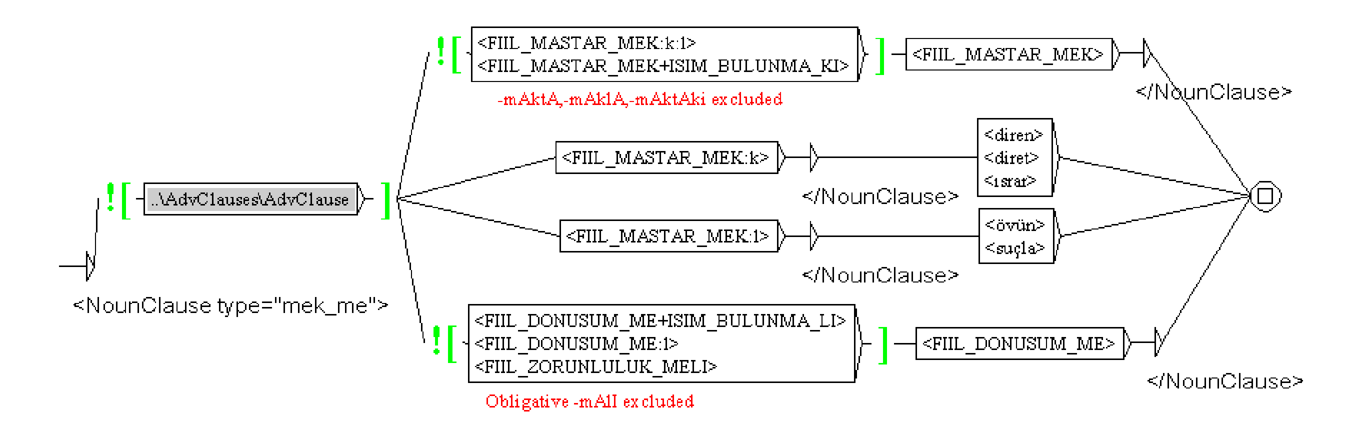


FIG. G.3 – Grammaire des propositions substantives *-mA/-mAk*

Annexe H

Texte avec étiquetage des subordonnées non finies

Ce qui suit est un extrait du corpus *RevOTAN* auquel a été appliqué le graphe Unitex des subordonnées non finies en mode *MERGE*. Les formes verbales repérées sont signalées par des balises XML.

{S}1999 yılında Kosova’daki etnik temizliği **<AdvClause sem="purpose"> durdurmak için </AdvClause>** **<RelClause> girişilen </RelClause>** 78 günlük hava kampanyası, ABD ile Müttefiklerin askeri yetenekleri arasındaki kritik “fay hattı”nı gözler önüne serdi.

{S} Bugün sayılar herkes tarafından biliniyor : hassas güdümlü mühimmatın yüzde 90’ı Amerikan avcı ve bombardıman uçakları tarafından kullanıldı ;

{S} Müttefiklerin sadece birkaç tanesi havada güvenli iletişimi bile güçlükle yürütebildi, ve NATO uçakları açık kanallardan **<NounClause type="mek_me"> haberleşmek </NounClause>** zorunda kaldı.

{S} Ayrıca NATO’nun “ **<NounClause type="mek_me"> karıştırma </NounClause>** ” yeteneğinin yüzde 100’ünü, havadan karaya **<NounClause type="mek_me"> gözetleme </NounClause>** yeteneğinin yüzde 90’ını, ve havada yakıt ikmal tankerlerinin yüzde 80’ini ABD sağladı.

{S} Bu büyük “fark” karşısında telaşa **<RelClause> kapılan </RelClause>** Lord Robertson, NATO’nun üç önceliğinin “yetenekler, yetenekler, yetenekler” **<NounClause type="mek_me"> olması </NounClause>** **<NounClause type="dik_ecek_nonSubj"> gerektiğini </NounClause>** **<NounClause type="mek_me"> tekrarlamaya </NounClause>** başladı.

{S}Ancak Kosova’da **<RelClause> alınan </RelClause>** dersler daha henüz **<AdvClause sem="time" tr="while"> özümsenirken </AdvClause>** NATO’nun stratejik ortamı 11 Eylül olaylarıyla sarsıldı.

{S} İttifak hızla **<AdvClause sem="manner"> davranarak </AdvClause>** tarihinde ilk defa **<AdvClause sem="manner"> olarak </AdvClause>** 5. maddeyi yürürlüğe soktu ve daha sonra da AWACS uçaklarını ABD şehirlerinin semalarında devriye görevine **<AdvClause**

sem="manner"> göndererek </AdvClause> toplu **<NounClause type="mek_me"> savunma </NounClause>** kavramına karşı **<RelClause> duyduğu </RelClause>** saygıyı gösterdi.
{S} Müttefik Dışişleri **<RelClause> Bakanları </RelClause>** Mayıs 2002'de Rejkavik'te **<RelClause> yaptıkları </RelClause>** toplantıda İttifak'ın güvenliğine yönelik tehditlerle mücadele **<AdvClause sem="purpose"> etmek için </AdvClause>** nereye gerekirse gitmekte kararlı **<NounClause type="dik_ecek_nonSubj"> olduğunu </NounClause>** resmen teyit ettiler.
{S} 2002 yılı boyunca NATO karargah personeli örgüt içinde geniş kapsamlı değişiklikler ve yeteneklerin **<NounClause type="mek_me"> arttırılması </NounClause>** ile ilgili planlar üzerinde titizlikle çalıştı ;
{S} bu planlar aynı yıl Kasım ayında **<RelClause> yapılan </RelClause>** Prag Zirvesi'nde İttifak liderleri tarafından onaylandı.
{S} Bunlar arasında NATO Mukabele Gücü'nün **<NounClause type="mek_me"> oluşturulması </NounClause>** , stratejik komutanlıkların yeniden **<NounClause type="mek_me"> düzenlenmesi </NounClause>** , ve Prag Yetenek Taahhütleri modernizasyon programlarının **<NounClause type="mek_me"> onaylanması </NounClause>** da vardı.
{S} Son **<AdvClause sem="manner"> olarak </AdvClause>** da İttifak yedi ülkeyi NATO'ya **<NounClause type="mek_me"> katılmaya </NounClause>** davet etti ve buna bağlı **<AdvClause sem="manner"> olarak </AdvClause>** Karargah yapısında gerçekleştirilecek reformları ve Kuzey Atlantik Konseyi'nin "26 üye" ile pürüzsüz biçimde **<AdvClause sem="purpose"> işleyebilmesi için </AdvClause>** **<RelClause> gereken </RelClause>** prosedürleri onayladı.
{S} **<RelClause> Geçen </RelClause>** iki buçuk yıl içinde bazı programlar geri kalmış olsa da Prag Gündemi'nin **<NounClause type="mek_me"> uygulanması </NounClause>** genelde olumlu oldu.
{S} İlk **<AdvClause sem="manner"> olarak </AdvClause>** Kuzey Atlantik Konseyi "26 üye" ile de **<NounClause type="dik_ecek_nonSubj"> çalışabileceğini </NounClause>** kanıtladı.
{S} Çek Cumhuriyeti'nin NATO nezdindeki Büyükelçisi Karel **<RelClause> Kovanda </RelClause>** 'nın Ekim 2003'te Almanya'daki Marshall Merkezinde **<RelClause> yaptığı </RelClause>** bir **<NounClause type="mek_me"> konuşmada </NounClause>** ifade **<AdvClause sem="manner"> ettiği gibi </AdvClause>** , "eğer belli bir çıkarları **<RelClause> olan </RelClause>** dört-beş Müttefik aralarında fikir birliği sağlarlarsa," toplam Müttefik sayısı 19 da olsa, 26 da olsa, genel fikir birliği "garanti edilmiş" **<NounClause type="mek_me"> demektir </NounClause>** .